

PENERAPAN HAAR CASCADE DALAM DETEKSI DAN PENGHITUNGAN JUMLAH WAJAH PADA GAMBAR DIGITAL

Sindy Susanti¹, Rihartanto², Supriadi³

^{1,2} Politeknik Negeri Samarinda, Jl. Dr. Cipto Mangunkusumo, samarinda 75131

Keywords:

HaarCascade
FaceDetection
OpenCV
Digital Image Processing

Correspondent

susantisindy768@gmail.com

Email:

Abstrak. Penelitian ini bertujuan untuk menerapkan metode Haar Cascade untuk deteksi dan penghitungan jumlah wajah pada gambar digital menggunakan Python dan OpenCV. Sistem bekerja melalui beberapa tahapan, meliputi input gambar, konversi ke grayscale, deteksi wajah, dan penghitungan jumlah wajah secara otomatis. Metode Haar Cascade dipilih karena memiliki kebutuhan komputasi yang ringan dan mampu melakukan pemrosesan secara real-time. Hasil penelitian menunjukkan bahwa sistem berhasil mendeteksi seluruh wajah sesuai dengan jumlah wajah sebenarnya yang terdapat pada gambar digital. Selain mudah diimplementasikan, metode ini juga efektif untuk aplikasi pengolahan citra digital sederhana. Penelitian ini diharapkan dapat menjadi referensi bagi pengembangan sistem deteksi wajah berbasis computer vision.



Copyright © 2026 The Author(s), Published by Jurnal Informatika dan Teknik Elektro Terapan. This is an open-access article distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Abstract. This study aims to implement the Haar Cascade method for face detection and face counting in digital images using Python and OpenCV. The system works through several stages, including image input, grayscale conversion, face detection, and automatic face counting. The Haar Cascade method was chosen because it has lightweight computational requirements and is capable of real-time processing. The results showed that the system successfully detected all faces according to the actual number of faces in the digital images. In addition to being easy to implement, this method is also effective for simple digital image processing applications. This research is expected to serve as a reference for the development of computer vision-based face detection systems.

1. PENDAHULUAN

Perkembangan teknologi pengolahan citra digital saat ini semakin pesat dan telah diterapkan dalam berbagai bidang, seperti keamanan, pendidikan, absensi otomatis, dan sistem pengawasan berbasis komputer. Salah satu teknologi yang banyak digunakan adalah deteksi wajah (face detection). Deteksi wajah merupakan proses untuk mengenali keberadaan wajah manusia pada sebuah citra atau video

menggunakan bantuan komputer [1]. Teknologi ini bertujuan untuk mempermudah sistem dalam mengidentifikasi objek wajah secara otomatis tanpa campur tangan manusia. Dalam implementasinya, diperlukan metode yang mampu bekerja secara cepat dan akurat agar dapat digunakan secara efektif pada berbagai kondisi citra digital. Salah satu metode yang populer dan sering digunakan adalah Haar Cascade yang diperkenalkan oleh Viola dan

Jones (2001) [13]. Metode ini dikenal memiliki kemampuan mendeteksi objek wajah dengan proses komputasi yang relatif cepat dan ringan sehingga cocok diterapkan pada aplikasi berbasis computer vision[2].

Metode Haar Cascade bekerja dengan cara mengenali pola fitur wajah menggunakan classifier yang telah dilatih sebelumnya melalui proses machine learning[1],[12]. Metode ini menggunakan fitur Haar berupa pola perbedaan intensitas warna terang dan gelap pada area tertentu yang umumnya terdapat pada wajah manusia, seperti area mata, hidung, dan pipi. Dalam proses implementasinya, sistem terlebih dahulu melakukan konversi citra berwarna menjadi grayscale agar proses pendeteksian dapat berjalan lebih cepat dan efisien. Penggunaan citra grayscale membantu mengurangi kompleksitas data warna sehingga classifier lebih mudah mengenali pola fitur wajah pada citra digital. Setelah proses grayscale selesai, sistem melakukan pemindaian pada seluruh area citra menggunakan fungsi `detectMultiScale()` untuk mencari objek yang memiliki karakteristik menyerupai wajah manusia. Penggunaan library OpenCV sangat membantu dalam implementasi metode Haar Cascade karena menyediakan berbagai fungsi pengolahan citra dan pendeteksian objek yang lengkap serta mudah digunakan. Menurut Bradski (2000), OpenCV merupakan library computer vision yang dirancang untuk mendukung pengolahan citra dan pendeteksian objek secara real-time[5],[7]. Dengan dukungan bahasa pemrograman Python yang memiliki sintaks sederhana dan mudah dipahami, metode Haar Cascade dapat diterapkan dengan proses pemrograman yang relatif mudah namun tetap mampu menghasilkan deteksi wajah yang cukup baik. Selain itu, metode ini juga memiliki proses komputasi yang ringan sehingga dapat dijalankan pada komputer dengan spesifikasi standar tanpa memerlukan perangkat keras khusus. Oleh karena itu, Haar Cascade masih sering digunakan dalam penelitian pengolahan citra digital maupun pengembangan aplikasi sederhana berbasis computer vision yang berfokus pada deteksi dan penghitungan jumlah wajah secara otomatis[3],[13].

Penelitian ini berfokus pada penerapan Haar Cascade dalam mendeteksi sekaligus menghitung jumlah wajah pada gambar digital.

Sistem yang dibangun bertujuan untuk mengetahui kemampuan metode Haar Cascade dalam mengenali dan menghitung jumlah wajah secara otomatis dengan tingkat akurasi yang baik. Proses pengujian dilakukan dengan membandingkan jumlah wajah asli pada gambar dengan jumlah wajah yang berhasil dideteksi oleh sistem. Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi seluruh wajah dengan jumlah deteksi yang sesuai dengan jumlah wajah sebenarnya. Hal tersebut menunjukkan bahwa metode Haar Cascade memiliki performa yang cukup baik dalam proses deteksi wajah, terutama pada kondisi pencahayaan yang memadai dan posisi wajah yang terlihat jelas.

Meskipun metode Haar Cascade memiliki keunggulan dalam hal kecepatan proses deteksi dan kemudahan implementasi pada berbagai aplikasi computer vision, metode ini tetap memiliki beberapa keterbatasan yang dapat memengaruhi tingkat akurasi sistem. Keberhasilan deteksi wajah sangat dipengaruhi oleh kondisi citra yang digunakan, seperti intensitas pencahayaan, kualitas gambar, posisi wajah, serta keberadaan objek lain yang dapat menutupi sebagian area wajah. Pada kondisi pencahayaan yang terlalu gelap atau terlalu terang, sistem sering mengalami kesulitan dalam mengenali pola fitur wajah karena perbedaan kontras yang kurang jelas. Selain itu, kualitas gambar yang rendah atau buram juga dapat menyebabkan fitur wajah tidak terbaca dengan baik oleh classifier sehingga proses deteksi menjadi kurang optimal. Faktor lain yang memengaruhi performa Haar Cascade adalah posisi wajah yang terlalu miring atau tidak menghadap langsung ke kamera. Metode ini cenderung bekerja lebih baik pada wajah dengan posisi frontal dibandingkan posisi samping atau sudut tertentu[3],[9]. Tidak hanya itu, keberadaan aksesoris seperti masker, kacamata, topi, maupun objek lain yang menutupi sebagian wajah juga dapat menyebabkan sistem gagal mendeteksi wajah secara sempurna.

Walaupun demikian, Haar Cascade masih menjadi salah satu metode deteksi wajah yang banyak digunakan karena memiliki proses komputasi yang ringan dan mampu bekerja

secara real-time pada berbagai perangkat komputer. Metode ini tidak memerlukan proses pelatihan model yang rumit maupun penggunaan perangkat keras dengan spesifikasi tinggi seperti metode deteksi wajah modern berbasis deep learning. Dibandingkan dengan metode modern yang membutuhkan GPU dan kapasitas memori besar, Haar Cascade lebih mudah diterapkan pada komputer dengan spesifikasi standar sehingga sangat cocok digunakan untuk kebutuhan pembelajaran, penelitian dasar, maupun pengembangan aplikasi sederhana berbasis computer vision[4],[6]. Selain itu, implementasi Haar Cascade menggunakan Python dan OpenCV juga relatif mudah dipahami karena OpenCV telah menyediakan berbagai fungsi pendukung deteksi objek yang lengkap. Kemudahan implementasi tersebut menjadi salah satu alasan metode Haar Cascade masih sering digunakan dalam pengolahan citra digital, terutama pada penelitian yang berfokus pada proses deteksi dan penghitungan jumlah wajah secara otomatis. Oleh karena itu, penelitian mengenai penerapan Haar Cascade dalam deteksi dan penghitungan jumlah wajah masih relevan untuk terus dikembangkan dan dianalisis lebih lanjut guna mengetahui tingkat efektivitas metode ini dalam berbagai kondisi citra digital, seperti perbedaan pencahayaan, posisi wajah, kualitas gambar, maupun gangguan objek lain pada citra. Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan teknologi deteksi wajah berbasis computer vision serta menjadi referensi bagi penelitian selanjutnya yang berkaitan dengan implementasi OpenCV dan pengolahan citra digital menggunakan bahasa pemrograman Python.

2. TINJAUAN PUSTAKA

2.1 Pengolahan Citra Digital

Pengolahan citra digital merupakan bidang ilmu yang mempelajari teknik untuk memperoleh, memperbaiki, menganalisis, dan menginterpretasikan informasi dari suatu citra menggunakan komputer. Pengolahan citra banyak diterapkan dalam berbagai bidang seperti keamanan, Kesehatan, Pendidikan, dan sistem pengawasan. Tujuan utama pengolahan citra adalah meningkatkan kualitas citra serta mengekstraksi informasi yang berguna untuk

proses analisis lebih lanjut[8]. Selain itu, pengolahan citra digital telah banyak diterapkan dalam berbagai penelitian, salah satunya untuk proses klasifikasi dan analisis citra berbasis kecerdasan buatan. Penelitian pada jurnal JITET menunjukkan bahwa teknik pengolahan citra digital dapat digunakan untuk mengklasifikasikan penyakit kulit wajah menggunakan metode *Convolutional Neural Network* [15].

2.2 Deteksi Wajah (Face Detection)

Deteksi Wajah merupakan proses untuk menemukan dan menentukan lokasi wajah manusia pada suatu gambar atau video. Teknologi ini menjadi bagian penting dalam bidang computer vision karena digunakan sebagai tahap awal pada sistem pengenalan wajah, absensi otomatis, sistem keamanan, dan pengawasan digital[3],[9]. Teknologi deteksi dan pengenalan wajah banyak diterapkan pada sistem presensi, keamanan, dan identifikasi pengguna secara otomatis.

2.3 Metode Haar Cascade

Haar Cascade merupakan metode yang diperkenalkan oleh Viola dan Jones pada tahun 2001. Metode ini bekerja dengan memanfaatkan fitur Haar-like untuk mengenali pola area tertentu. Haar Cascade menggunakan proses klasifikasi bertingkat (*Cascade Classifier*) sehingga mampu melakukan deteksi objek secara cepat dan efisien. Metode ini banyak digunakan dalam deteksi wajah karena memiliki kebutuhan komputasi yang relatif ringan dan dapat diterapkan secara real-time[2],[13].

2.4 OpenCV

OpenCV(*Open Source Computer Vision Library*) merupakan pustaka pemrograman yang menyediakan berbagai fungsi pengolahan citra dan komputer vision. OpenCV dirancang untuk mendukung pengolahan citra dan pendeteksian objek secara real-time sehingga banyak digunakan dalam penelitian maupun pengembangan aplikasi komputer vision[5],[7].

2.5 Python

Python adalah bahasa pemrograman tingkat tinggi yang banyak digunakan dalam bidang pengolahan data, kecerdasan buatan, machine learning, dan *computer vision*. Python memiliki sintaks yang sederhana dan mudah dipahami sehingga memudahkan proses pengembangan aplikasi. Dalam penelitian ini,

Python digunakan sebagai bahasa pemrograman utama untuk mengimplementasikan metode Haar Cascade dengan bantuan library OpenCV[7],[10].

2.6 Grayscale

Grayscale adalah proses citra berwarna menjadi citra abu-abu, konversi *grayscale* dilakukan untuk mengurangi kompleksitas data warna sehingga proses pengolahan citra menjadi lebih cepat dan efisien. Pada metode Haar Cascade, citra *grayscale* digunakan sebagai proses pendeteksian objek lebih efektif dibandingkan menggunakan citra berwarna[8].

3. METODE PENELITIAN

Metode penelitian yang digunakan pada penelitian ini adalah metode eksperimen dengan menerapkan algoritma Haar Cascade untuk mendeteksi dan menghitung jumlah wajah pada citra digital menggunakan bahasa pemrograman python dan library OpenCV. Penelitian dilakukan secara bertahap mulai dari pengumpulan data citra, perancangan sistem, implementasi algoritma, Haar Cascade, hingga pengujian hasil deteksi wajah. Tujuan penelitian ini adalah untuk mengetahui kemampuan metode Haar Cascade dalam mendeteksi wajah dan menghitung jumlah wajah yang terdapat pada gambar digital secara otomatis.

3.1 Analisis Sistem

Analisis sistem dilakukan untuk mengetahui kebutuhan sistem yang akan dibangun serta proses yang diperlukan dalam mendeteksi dan menghitung jumlah wajah pada citra digital. Sistem dirancang untuk menerima masukan berupa gambar digital yang mengandung satu atau lebih wajah manusia. Sistem melakukan proses pengolahan citra menggunakan metode Haar Cascade untuk mengenali objek wajah dan menghitung jumlah wajah yang berhasil dideteksi.

3.2 Analisis kebutuhan Sistem

Analisis kebutuhan dilakukan untuk memahami kebutuhan sistem yang akan dibangun sebelum dilakukan perancangan dan implementasi, Penelitian ini menggunakan perangkat keras berupa komputer/laptop dengan spesifikasi yang mampu menjalankan Python 3.10 dan OpenCV. Metode Haar Cascade tidak membutuhkan spesifikasi perangkat yang tinggi karena proses

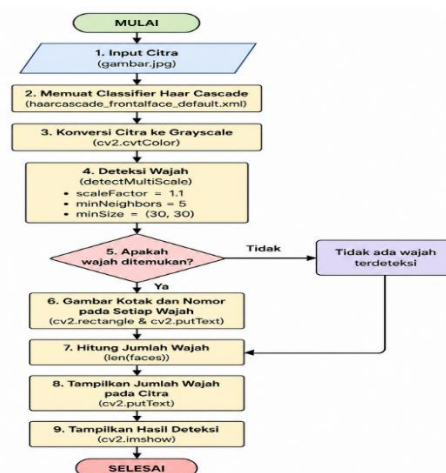
komputasinya relative ringan dibandingkan metode deteksi wajah berbasis deep learning.

Selanjutnya perangkat lunak yang digunakan yaitu Python 3.10 sebagai bahasa pemrograman, OpenCV sebagai library pengolahan citra digital, Visual Studio Code sebagai editor pemrograman, serta file *classifier Haar Cascade XML* yang digunakan sebagai model deteksi wajah.

3.3 Arsitektur Sistem

Arsitektur Sistem menggambarkan hubungan antara komponen-komponen yang digunakan dalam proses deteksi wajah, sistem terdiri dari beberapa komponen utama yaitu gambar digital sebagai input, proses pengolahan citra, algoritma Haar Cascade, dan hasil deteksi wajah sebagai output.

Tahap awal sistem menerima gambar digital sebagai masukan, gambar kemudian diproses menggunakan OpenCV untuk dilakukan konversi ke grayscale. Setelah itu sistem memuat classifier Haar Cascade dan melakukan proses deteksi wajah. Hasil deteksi digunakan untuk menentukan jumlah wajah yang terdapat pada gambar, selanjutnya sistem menampilkan gambar yang telah diberi kotak penanda serta informasi jumlah wajah yang berhasil dideteksi.



Gambar 1 Arsitektur Sistem

3.4 Metode Haar Cascade

Penelitian ini menggunakan metode Haar Cascade yang diperkenalkan oleh Viola dan Jones, metode ini bekerja dengan memanfaatkan fitur Haar-like untuk mengenali pola tertentu pada objek wajah manusia. Metode Haar Cascade dipilih karena memiliki proses komputasi yang ringan, mudah

diimplementasikan menggunakan OpenCV, serta mampu melakukan deteksi wajah secara real-time pada perangkat dengan spesifikasi standar[2],[12],[13].

3.5 Implementasi Sistem

Implementasi system menggunakan bahasa pemrograman Python 3.10 dan library OpenCV, Program diawali dengan memanggil library cv2 yang berfungsi sebagai pustaka utama dalam pengolahan citra digital. Setelah itu, sistem memuat classifier Haar Cascade XML yang digunakan untuk mengenali pola wajah manusia. Implementasi sistem menggunakan Python dan OpenCV banyak diterapkan dalam pengembangan sistem deteksi wajah otomatis maupun sistem absensi berbasis pengenalan wajah [7], [10]. Tahap berikutnya adalah membaca citra yang akan diproses menggunakan fungsi *cv2.imread()*. Citra kemudian dikonversi menjadi grayscale menggunakan fungsi *cv2.cvtColor()* karena metode Haar Cascade bekerja lebih efektif pada citra abu-abu dibandingkan citra berwarna. Proses grayscale membantu sistem dalam mengurangi kompleksitas data sehingga proses deteksi dapat dilakukan dengan lebih cepat dan efisien.

Setelah proses konversi selesai, sistem melakukan pendeteksian wajah menggunakan fungsi *detectMultiScale()*. Fungsi ini bekerja dengan memindai seluruh area citra menggunakan beberapa ukuran skala untuk menemukan pola fitur wajah manusia. Parameter *scaleFactor* digunakan untuk menentukan skala pengecilan gambar pada setiap proses pemindaian, sedangkan *minNeighbors* berfungsi untuk meningkatkan tingkat akurasi deteksi dengan mengurangi kemungkinan kesalahan pembacaan objek selain wajah. Parameter *minSize* digunakan untuk menentukan ukuran minimum objek yang dapat dideteksi sebagai wajah. Ketika sistem berhasil mendeteksi wajah, program akan membuat kotak berwarna hijau pada area wajah menggunakan fungsi *cv2.rectangle()* serta memberikan nomor identitas pada setiap wajah yang berhasil dikenali menggunakan fungsi *cv2.putText()*.

```

1 import cv2
2 # Memanggil file Haar Cascade XML
3 face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
4 # Membaca gambar input
5 image = cv2.imread('foto_1_orang.jpg')
6 # Validasi jika gambar tidak terbaca
7 if image is None:
8     print("Gambar tidak ditemukan. Periksa nama file dan lokasi gambar.")
9     exit()
10 # Mengubah gambar ke grayscale
11 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
12 # Melakukan deteksi wajah menggunakan Haar Cascade
13 faces = face_cascade.detectMultiScale(
14     gray,
15     scaleFactor=1.1,
16     minNeighbors=5,
17     minSize=(30, 30)
18 )
19 # Menggambar kotak dan nomor pada setiap wajah yang terdeteksi
20 for i, (x, y, w, h) in enumerate(faces, start=1):
21     cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
22     cv2.putText(
23         image,
24         str(i),
25         (x, y - 10),
26         cv2.FONT_HERSHEY_SIMPLEX,
27         0.8,
28         (0, 255, 0),
29         2
30     )
31 # Menghitung jumlah wajah terdeteksi
32 jumlah_wajah = len(faces)
33 # Menampilkan teks jumlah wajah pada gambar
34 cv2.putText(
35     image,
36     ("Jumlah Wajah Terdeteksi: {}".format(jumlah_wajah)),
37     (20, 40),

```

Gambar 2. Kode Program 1

4. HASIL DAN PEMBAHASAN

4.1 Hasil Implementasi Sistem

Berdasarkan hasil implementasi sistem deteksi wajah menggunakan metode Haar Cascade, citra digital yang diuji berhasil diproses melalui beberapa tahapan, yaitu input citra, konversi ke grayscale, pemanggilan classifier Haar Cascade, proses deteksi wajah, pemberian kotak penanda, serta penghitungan jumlah wajah. Pada hasil sebelum deteksi, citra hanya menampilkan objek wajah tanpa adanya informasi tambahan dari sistem. Setelah dilakukan proses deteksi, sistem memberikan kotak berwarna hijau pada setiap wajah yang berhasil dikenali dan menampilkan jumlah wajah secara otomatis. Hasil pengujian menunjukkan bahwa sistem mendeteksi sebanyak 4 wajah, sesuai dengan jumlah wajah asli yang terdapat pada citra. Dengan demikian, hasil deteksi menunjukkan kesesuaian antara jumlah wajah sebenarnya dan jumlah wajah yang berhasil dikenali oleh sistem.

Hasil tersebut menunjukkan bahwa metode Haar Cascade mampu bekerja dengan baik pada citra yang memiliki kondisi wajah cukup jelas, pencahayaan memadai, dan posisi wajah yang sebagian besar menghadap ke kamera. Proses deteksi dilakukan dengan fungsi *detectMultiScale()* yang memindai citra dalam berbagai skala untuk menemukan pola fitur wajah. Parameter yang digunakan, seperti *scaleFactor*, *minNeighbors*, dan *minSize*, berperan penting dalam menentukan keberhasilan sistem saat mengenali wajah. Pada hasil deteksi, setiap wajah diberi nomor urut sehingga proses penghitungan

menjadi lebih mudah diamati. Sistem tidak hanya menandai area wajah, tetapi juga menghitung total wajah yang terdeteksi menggunakan fungsi *len(faces)*. Hasil akhir berupa tampilan jumlah wajah terdeteksi yang sesuai dengan jumlah wajah asli, yaitu 4 wajah.



Gambar 3. Hasil Sebelum Deteksi



Gambar 4. Hasil Sesudah Deteksi

4.2 Hasil Pengujian Sistem

Pengujian dilakukan untuk mengetahui kemampuan metode Haar Cascade dalam mendeteksi dan menghitung jumlah wajah pada citra digital.

4.2.1 Tabel Pengujian Deteksi wajah

Tabel 1. Hasil pengujian Deteksi Wajah

Jumlah Wajah Aktual	Jumlah Wajah Terdeteksi	Akurasi
1	1	100%
2	2	100%
3	3	100%
4	4	100%

Pada proses pengujian, jumlah wajah hasil deteksi menunjukkan hasil yang sama dengan jumlah wajah asli pada citra digital. Hal tersebut menunjukkan bahwa metode Haar Cascade mampu bekerja dengan baik dalam mengenali pola fitur wajah manusia pada kondisi citra

yang memiliki pencahayaan cukup dan kualitas gambar yang jelas. Keberhasilan deteksi dipengaruhi oleh penggunaan parameter yang tepat pada fungsi *detectMultiScale()*, seperti *scaleFactor*, *minNeighbors*, dan *minSize*, sehingga sistem dapat mengurangi kesalahan pembacaan objek selain wajah. Dengan adanya proses deteksi otomatis tersebut, metode Haar Cascade mampu membantu proses pengolahan citra digital menjadi lebih efektif dan efisien dalam melakukan identifikasi serta penghitungan jumlah wajah.

4.2.2 Tabel Pengaruh kondisi Citra Terhadap Hasil Deteksi

Tabel 2. Pengaruh kondisi Citra

Kondisi Citra	Hasil Deteksi
Pencahayaan cerah	Sangat baik
Pencahayaan redup	Baik
Wajah frontal	Sangat baik
Wajah miring	Kurang Optimal
Citra buram	Kurang Optimal

Hasil pengujian menunjukkan bahwa jumlah wajah hasil deteksi sesuai dengan jumlah wajah asli yang terdapat pada citra. Hal tersebut menunjukkan bahwa metode Haar Cascade mampu bekerja secara efektif dalam mengenali pola fitur wajah manusia. Keberhasilan sistem dipengaruhi oleh kondisi citra yang mendukung, seperti pencahayaan yang cukup, kualitas gambar yang jelas, dan posisi wajah yang tidak terlalu miring. Selain itu, proses konversi grayscale juga membantu sistem mengurangi kompleksitas warna sehingga proses pendeteksian menjadi lebih cepat dan efisien.

4.3 Pembahasan

Berdasarkan hasil pengujian yang dilakukan, metode Haar Cascade menunjukkan performa yang cukup baik dalam mendeteksi dan menghitung jumlah wajah pada citra digital. Sistem mampu menampilkan perbedaan yang jelas antara kondisi sebelum deteksi dan sesudah deteksi. Sebelum proses deteksi dilakukan, citra hanya menampilkan objek wajah tanpa adanya informasi tambahan dari sistem. Setelah proses deteksi dijalankan, sistem memberikan kotak penanda pada setiap wajah yang berhasil dikenali serta menampilkan jumlah wajah secara otomatis. Hasil pengujian menunjukkan bahwa jumlah wajah hasil deteksi sesuai dengan jumlah wajah

asli yang terdapat pada citra. Hal tersebut menunjukkan bahwa metode Haar Cascade mampu bekerja secara efektif dalam mengenali pola fitur wajah manusia. Hasil penelitian ini sejalan dengan penelitian sebelumnya yang menunjukkan bahwa teknologi deteksi dan pengenalan wajah dapat diterapkan secara efektif pada berbagai sistem berbasis *computer vision* [12]. Keberhasilan sistem dipengaruhi oleh kondisi citra yang mendukung, seperti pencahayaan yang cukup, kualitas gambar yang jelas, dan posisi wajah yang tidak terlalu miring. Selain itu, proses konversi grayscale juga membantu sistem mengurangi kompleksitas warna sehingga proses pendeteksian menjadi lebih cepat dan efisien[1],[8],[12].

Meskipun hasil deteksi menunjukkan performa yang baik, metode Haar Cascade tetap memiliki beberapa keterbatasan yang dapat memengaruhi tingkat akurasi sistem. Pada kondisi tertentu, seperti pencahayaan rendah, citra buram, wajah tertutup sebagian objek, atau posisi wajah terlalu miring, sistem dapat mengalami kesalahan deteksi atau bahkan gagal mengenali wajah. Hal ini terjadi karena metode Haar Cascade bekerja berdasarkan pola fitur sederhana sehingga sangat bergantung pada kualitas citra yang digunakan. Selain itu, metode ini lebih optimal dalam mendeteksi wajah dengan posisi frontal dibandingkan posisi samping atau sudut tertentu[3],[9]. Walaupun demikian, metode Haar Cascade masih banyak digunakan karena memiliki keunggulan dalam kecepatan proses dan kemudahan implementasi dibandingkan metode deteksi modern yang memerlukan spesifikasi perangkat keras lebih tinggi[4],[6],[13]. Dengan menggunakan Python 3.10, OpenCV, dan classifier XML, sistem deteksi wajah dapat dibangun dengan cukup sederhana namun tetap mampu memberikan hasil yang baik dalam proses penghitungan jumlah wajah secara otomatis[6],[13]. Oleh karena itu, metode Haar Cascade masih relevan digunakan sebagai media pembelajaran dan dasar pengembangan sistem *computer vision* berbasis pengolahan citra digital[2],[13].

Selain menunjukkan tingkat akurasi yang tinggi pada citra uji, hasil penelitian ini juga membuktikan bahwa metode Haar Cascade

masih relevan digunakan pada aplikasi pengolahan citra digital sederhana. Tingkat akurasi 100% yang diperoleh pada pengujian menunjukkan bahwa classifier mampu mengenali pola wajah dengan baik pada kondisi pencahayaan yang memadai. Hasil ini sejalan dengan penelitian Nugroho dan Sela [1] serta Ranjini dkk. [12] yang menyatakan bahwa Haar Cascade memiliki performa yang baik untuk deteksi wajah real-time. Meskipun demikian, pengujian pada penelitian ini masih terbatas pada jumlah sampel yang relatif sedikit sehingga diperlukan pengujian lebih lanjut menggunakan variasi citra yang lebih beragam. Dengan demikian, penelitian selanjutnya dapat melakukan pengujian pada kondisi pencahayaan ekstrem, wajah dengan orientasi berbeda, maupun citra dengan kualitas rendah untuk memperoleh evaluasi performa yang lebih komprehensif.

5. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, penerapan metode Haar Cascade dalam deteksi dan penghitungan jumlah wajah pada citra digital dapat berjalan dengan baik sesuai tujuan penelitian. Sistem yang dibangun menggunakan Python 3.10, OpenCV, dan classifier Haar Cascade XML mampu mendeteksi wajah secara otomatis melalui beberapa tahapan, mulai dari input citra, konversi grayscale, proses deteksi wajah, hingga penghitungan jumlah wajah yang berhasil dikenali. Hasil pengujian menunjukkan bahwa sistem mampu memberikan kotak penanda pada setiap wajah yang terdeteksi dan menampilkan jumlah wajah secara otomatis sesuai dengan jumlah wajah sebenarnya pada citra digital. Keberhasilan tersebut menunjukkan bahwa metode Haar Cascade cukup efektif digunakan dalam proses pengolahan citra digital, khususnya untuk mendeteksi dan menghitung jumlah wajah secara otomatis. Selain memiliki proses komputasi yang ringan, metode ini juga mudah diimplementasikan sehingga cocok digunakan sebagai media pembelajaran maupun pengembangan aplikasi sederhana berbasis *computer vision*.

Hasil penelitian juga menunjukkan bahwa tingkat keberhasilan deteksi sangat

dipengaruhi oleh kondisi citra yang digunakan. Faktor seperti pencahayaan yang memadai, posisi wajah yang jelas menghadap kamera, dan kualitas gambar yang baik membantu sistem dalam mengenali pola fitur wajah dengan lebih optimal. Proses konversi citra menjadi grayscale juga membantu mengurangi kompleksitas data warna sehingga proses deteksi dapat dilakukan dengan lebih cepat dan efisien. Namun demikian, metode Haar Cascade masih memiliki beberapa keterbatasan. Sistem dapat mengalami penurunan performa apabila citra memiliki kualitas rendah, pencahayaan kurang baik, posisi wajah terlalu miring, atau terdapat objek yang menutupi sebagian area wajah. Meskipun begitu, metode Haar Cascade tetap menjadi salah satu metode deteksi wajah yang masih relevan digunakan karena memiliki kecepatan proses yang baik dan tidak memerlukan perangkat keras dengan spesifikasi tinggi seperti metode deteksi modern berbasis deep learning.

Secara keseluruhan, penelitian ini membuktikan bahwa metode Haar Cascade dapat digunakan sebagai solusi sederhana dalam proses deteksi dan penghitungan jumlah wajah pada citra digital. Implementasi menggunakan Python dan OpenCV memberikan kemudahan dalam proses pengembangan sistem sehingga dapat diterapkan pada berbagai kebutuhan pengolahan citra digital berbasis computer vision. Penelitian ini diharapkan dapat menjadi referensi bagi pengembangan penelitian selanjutnya yang berkaitan dengan deteksi wajah otomatis maupun pengembangan aplikasi *computer vision* lainnya. Untuk penelitian berikutnya, sistem dapat dikembangkan lebih lanjut dengan menambahkan metode deteksi wajah yang lebih modern agar mampu bekerja lebih optimal pada berbagai kondisi citra yang kompleks, seperti pencahayaan rendah, posisi wajah tidak frontal, maupun objek wajah yang tertutup sebagian.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih sebesar-besarnya kepada:

- Rihartanto, S.T., M.Si dan Supriadi, SST. MT. selaku dosen pembimbing yang telah

memberikan bimbingan, arahan, dan masukan berharga selama penelitian ini.

- Kedua orang tua tercinta atas segala do'a, dukungan, motivasi, serta pengorbanan yang tiada henti. Semoga segala kebaikan, cinta, dan pengorbanan yang telah diberikan mendapatkan balasan yang berlimpah dari Tuhan Yang Maha Esa.
- Terima kasih juga untuk teman-teman saya yang telah memberi semangat dan dukungan dalam menjalankan jurnal ini.

DAFTAR PUSTAKA

- [1] F. T. Nugroho dan E. I. Sela, "Deteksi Citra Wajah Menggunakan Algoritma Haar Cascade Classifier," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 1, pp. 37–44, Jan. 2024. doi: 10.57152/malcom.v4i1.988.
- [2] D. M. Abdilhussien dan L. J. Saud, "An Evaluation Study of Face Detection by Viola-Jones Algorithm," *International Journal of Health Sciences*, vol. 6, no. S8, pp. 4174–4182, 2022. doi: 10.53730/ijhs.v6nS8.13127.
- [3] K. Hasan, S. Ahsan, Abdullah-Al-Mamun, S. H. S. Newaz, dan G. M. Lee, "Human Face Detection Techniques: A Comprehensive Review and Future Research Directions," *Electronics*, vol. 10, no. 19, 2021. doi: 10.3390/electronics10192354.
- [4] D. Mamieva, A. B. Abdusalomov, M. Mukhiddinov, dan T. K. Whangbo, "Improved Face Detection Method via Learning Small Faces on Hard Images Based on a Deep Learning Approach," *Sensors*, vol. 23, no. 1, p. 502, Jan. 2023. doi: 10.3390/s23010502.
- [5] R. Jailani et al., "Real-Time Head Pose Estimation Using Haar Cascade and OpenCV," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 3, pp. 130–140, 2024.
- [6] M. M. Abid, T. Mahmood, R. Ashraf, C. M. N. Faisal, H. Ahmad, dan A. A. Niaz, "Computationally Intelligent Real-Time Security Surveillance System in the Education Sector Using Deep Learning," *PLOS ONE*, vol. 19, no. 7, p. e0301908, Jul. 2024. doi: 10.1371/journal.pone.0301908.
- [7] A. Sain, S. Dutta, R. Saha, dan U. Mandal, "Automated Facial Recognition Based Attendance System Using OpenCV in Python," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*,

- vol. 9, no. 6, pp. 105–111, Nov.–Des. 2023. doi: 10.32628/CSEIT2390617.
- [8] Sunardi, A. Yudhana, dan S. A. Wijaya, "Application of Median and Mean Filtering Methods for Optimizing Face Detection in Digital Photo," *Revue d'Intelligence Artificielle*, vol. 37, no. 2, pp. 291–297, Apr. 2023. doi: 10.18280/ria.370206.
- [9] T. H. Obaida, A. S. Jamil, dan N. F. Hassan, "Real-Time Face Detection in Digital Video Based on Viola-Jones Supported by Convolutional Neural Networks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 3, 2022. doi: 10.11591/ijece.v12i3.
- [10] O. Khalkar, T. Bhosale, S. Yadav, T. Galande, dan A. Kadam, "Automatic Attendance System Using Face Detection and Machine Learning," *International Journal of Novel Research and Development (IJNRD)*, vol. 9, no. 2, pp. 273–277, 2024.
- [11] A. Dewi dan E. Yulianto, "Evaluasi Performa Metode Deteksi Objek Berbasis Python Menggunakan Metrik IoU dan Precision," *Jurnal Sistem Informasi dan Informatika*, pp. 23–31, 2023.
- [12] M. M. Ranjini et al., "Haar Cascade Classifier-Based Real-Time Face Recognition and Face Detection," dalam *Proc. 4th International Conference on Smart Electronics and Communication (ICOSEC)*, 2023.
- [13] V. V. Arlazarov, J. S. Voysyat, D. P. Matalov, D. P. Nikolaev, dan S. A. Usilin, "Evolution of the Viola-Jones Object Detection Method: A Survey," *Bulletin of the South Ural State University, Series: Mathematical Modelling, Programming and Computer Software*, vol. 14, no. 4, 2021. doi: 10.14529/mmp210401.
- [14] Sunardi, S. Fadlil, dan D. Prayogi, "Sistem Pengenalan Wajah pada Keamanan Ruang Berbasis Convolutional Neural Network," *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, vol. 6, no. 2, pp. 636–647, 2022.
- [15] R. A. B. Kusuma, B. Irawan, dan A. Khamid, "Klasifikasi Penyakit Kulit Wajah Menggunakan Convolutional Neural Network EfficientNet-B3," *Jurnal Informatika dan Teknik Elektro Terapan (JITET)*, vol. 14, no. 1, 2026. doi: 10.23960/jitet.v14i1.8721.