

Terakreditasi SINTA Peringkat 3

Surat Keputusan Direktur Jenderal Pendidikan Tinggi, Riset, dan Teknologi Nomor 225/E/KPT/2022 masa berlaku mulai Vol.7 No. 1 tahun 2022 s.d Vol. 11 No. 2 tahun 2026

Terbit online pada laman web jurnal:
<http://publishing-widyagama.ac.id/ejournal-v2/index.php/jointecs>



Vol. 8 No. 2 (2023) 49 -56

JOINTECS

(Journal of Information Technology and Computer Science)

e-ISSN:2541-6448

p-ISSN:2541-3619

Deteksi Mata di Video Smartphone Menggunakan Mediapipe Python

Muhammad Furqan Rasyid¹, Muhammad Syukri Mustafa², Andi Asvin Mahersatillah Suradi³,
Muhammad Rizal⁴, Mushaf⁵, Arham Arifin⁶

^{1, 2, 5} Program Studi Manajemen Informatika, Universitas Dipa Makassar

^{4, 3, 6} Program Studi Teknik Informatika, Universitas Dipa Makassar

¹ muhammad.furqan@undipa.ac.id, ² syukri@undipa.ac.id, ³ andiasvin@undipa.ac.id,

⁴ muhammad.rizal@undipa.ac.id, ⁵ mushafsyafar@undipa.ac.id, ⁶ arhamarifin@undipa.ac.id

Abstract

Eye detection technology is used to recognize and analyze unique features of a person's eyes as a way to identify or authenticate their identity. This technology can be used in various applications such as pattern recognition, biometric systems, surveillance systems, and others. Most applications require precision in eye detection, so a fast and reliable eye detection method is needed. In this research, an eye detection method is proposed using the Python OpenCV and MediaPipe libraries, which offer better accuracy compared to existing solutions. Both libraries are implemented in the Python programming language, which is popular among software developers for its ability in object-oriented programming, easy data manipulation and processing, and availability of libraries and modules in various fields such as artificial intelligence. The system was tested using videos captured using a smartphone. Although the videos were captured under suboptimal conditions, such as imperfect lighting, testing was conducted on 56 videos that had relatively good quality and lasted about 5-10 seconds. The results obtained showed an accuracy rate of 100%. Additionally, the system can distinguish between open and closed eye conditions, which will facilitate further research in detecting eye blinks. In conclusion, the model created can detect eyes with a very high accuracy rate.

Keywords: eyes detection; mediapipe; opencv; python; smartphone.

Abstrak

Teknologi deteksi mata digunakan untuk mengenali dan menganalisis fitur-fitur unik pada mata seseorang sebagai cara untuk mengidentifikasi atau mengautentikasi identitas seseorang. Teknologi ini dapat digunakan dalam berbagai aplikasi, seperti pengenalan pola, sistem biometrik, sistem pengawasan, dan lainnya. Kebanyakan aplikasi memerlukan ketepatan dalam mendeteksi mata, sehingga diperlukan metode deteksi mata yang cepat dan andal. Dalam penelitian ini, diajukan metode deteksi mata yang menggunakan library Python OpenCV dan MediaPipe, yang menawarkan akurasi yang lebih baik dibandingkan solusi yang sudah ada. Kedua pustaka tersebut diimplementasikan dalam bahasa pemrograman Python, yang populer di kalangan pengembang perangkat lunak karena kemampuan pemrograman berorientasi objek, kemampuan untuk memanipulasi dan memproses data dengan mudah, serta pustaka dan modul yang tersedia dalam berbagai bidang seperti kecerdasan buatan. Pengujian sistem dilakukan dengan menggunakan video yang diambil menggunakan telepon pintar. Meskipun video diambil dalam kondisi kurang optimal, yaitu dengan pencahayaan yang tidak sempurna, pengujian dilakukan pada 56 video yang memiliki kualitas cukup baik dengan durasi sekitar 5-10 detik. Hasil yang diperoleh menunjukkan tingkat akurasi yang mencapai 100%. Selain itu, sistem yang dibuat mampu membedakan antara kondisi mata terbuka dan tertutup, yang akan memudahkan penelitian selanjutnya dalam mendeteksi kedipan mata. Kesimpulan yang dapat diambil adalah model yang telah dibuat mampu mendeteksi mata dengan tingkat akurasi yang sangat tinggi.

Kata kunci: deteksi mata; mediapipe; opencv; python; telepon pintar.



1. Pendahuluan

Ponsel pintar atau smartphone merupakan jenis komputer portable yang menggabungkan daya komputasi dengan fungsionalitas telepon. Ponsel pintar berbeda dari telepon pada umumnya karena memiliki perangkat keras yang lebih kuat dan sistem operasi seluler yang komprehensif yang memungkinkan akses ke perangkat lunak yang lebih luas, penelusuran web melalui broadband seluler, dan fungsionalitas multimedia (seperti musik, video, kamera, dan game). Smartphone dengan kamera yang baik dan kemampuan untuk merekam video definisi tinggi sudah tersedia secara luas. Namun Video yang dihasilkan menggunakan smartphone memiliki kualitas yang kurang baik bila dibandingkan dengan kamera profesional.

Disini lain, deteksi mata adalah teknologi penting di bidang visi komputer dan pembelajaran mesin. Ini mengacu pada proses mengidentifikasi atau menemukan mata dalam gambar atau video, yang digunakan dalam berbagai aplikasi seperti pengenalan wajah, pelacakan tatapan, dan interaksi manusia-komputer. Teknologi deteksi mata biasanya menggunakan analisis gambar, pembelajaran mesin, dan algoritma visi komputer untuk mendeteksi dan menemukan mata dalam sebuah gambar. Beberapa fitur utama yang digunakan untuk mendeteksi mata antara lain bentuk mata, posisi relatif mata terhadap fitur wajah lainnya, serta tekstur dan warna mata. Ada berbagai algoritma dan teknik yang dikembangkan untuk deteksi mata, seperti Haar Cascades, algoritma Viola-Jones, HOG (Histograms of Oriented Gradients) dan CNN (Convolutional Neural Network). Beberapa pustaka dan kerangka kerja yang menyediakan fungsionalitas deteksi Mata adalah OpenCV, dlib, dan MediaPipe. Bahkan, di beberapa penelitian lainnya, pustaka OpenCv sering kali digunakan untuk segmentasi warna HSV[1].

Sejumlah Penelitian telah dilakukan mengenai deteksi mata. Penelitian yang dilakukan oleh [2] mendeteksi mata menggunakan teknologi pemrosesan citra termal. Metode yang mereka usulkan mencapai akurasi yang tinggi dan respons cepat dalam kondisi nyata tanpa kerumitan komputasi dan persyaratan kumpulan data besar yang terkait dengan jaringan saraf tiruan. Penelitian lain yang dilakukan oleh [3] mendeteksi mata dengan menggunakan Dlib 68. Lebih lanjut, penelitian yang mereka lakukan untuk mendeteksi kedipan mata.

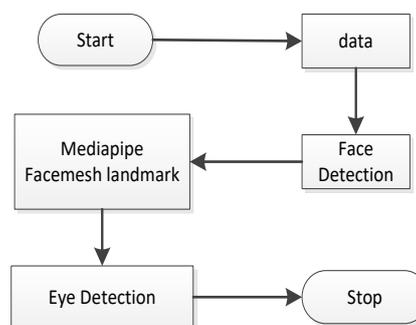
MediaPipe adalah sebuah kerangka kerja (framework) yang digunakan untuk membangun jaringan pembelajaran mesin untuk memproses data deret waktu, termasuk video. Penggunaan MediaPipe dalam beberapa penelitian juga telah menunjukkan hasil yang menjanjikan. Sebagai contoh, penelitian oleh [4] menggunakan pustaka MediaPipe untuk melacak pergerakan tangan dengan hasil yang baik menggunakan GPU lite. Penelitian lainnya, seperti yang dilakukan

oleh [5] menggunakan library mediapipe untuk menemukan dan menganalisis pipeline dalam komputasi. Selain itu, MediaPipe juga telah digunakan dalam penelitian untuk melacak pergerakan tangan pasien penderita Parkinson oleh [6], dengan tingkat kesalahan yang rendah.

Namun, meskipun MediaPipe telah menunjukkan potensi yang menjanjikan dalam deteksi dan pelacakan anggota tubuh manusia, penggunaan MediaPipe untuk mendeteksi mata dalam video yang diambil menggunakan smartphone masih terbatas. Oleh karena itu, penelitian ini akan difokuskan pada evaluasi akurasi MediaPipe dalam mendeteksi mata dalam video dengan kualitas rendah yang dihasilkan oleh smartphone. Penelitian ini bertujuan untuk memberikan pemahaman yang lebih baik tentang potensi penggunaan MediaPipe dalam deteksi mata pada video dengan kualitas rendah. Selain itu, penelitian ini juga akan memperluas pemahaman kita tentang keterbatasan dan potensi pengembangan lebih lanjut dalam deteksi mata menggunakan smartphone. Diharapkan hasil penelitian ini dapat memberikan kontribusi signifikan bagi pengembangan teknologi deteksi mata dalam konteks penggunaan smartphone. Dengan pemahaman yang lebih baik tentang kemampuan dan keterbatasan deteksi mata menggunakan MediaPipe pada video dengan kualitas rendah, kita dapat melihat potensi pengembangan lebih lanjut untuk meningkatkan akurasi dan performa deteksi mata menggunakan smartphone.

2. Metode Penelitian

Metode penelitian merupakan prosedur khusus yang digunakan untuk mengumpulkan dan menganalisis data dalam suatu penelitian. Pengembangan metode penelitian menjadi bagian integral dari desain penelitian. Salah satu fungsi penting dari metode penelitian adalah menentukan pendekatan yang tepat dalam pengumpulan data. Dengan menggunakan metode penelitian yang sesuai, peneliti dapat meminimalkan kesalahan dalam pengumpulan data serta memastikan validitas dan reliabilitas data yang diperoleh. Gambaran umum mengenai metode penelitian yang digunakan dapat dilihat pada Gambar 1. Penjelasan selengkapnya mengenai metode yang diusulkan akan dibahas lebih lanjut pada sub-bab berikutnya (2.1, 2.2, 2.3, 2.4).



Gambar 1. Metode Penelitian

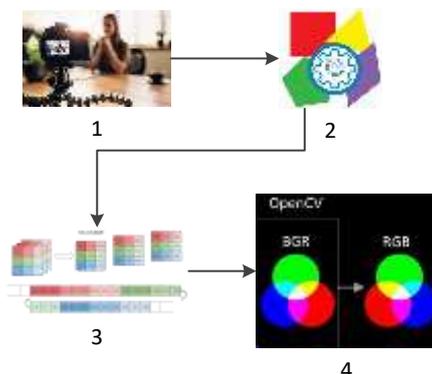
2.1. Pengumpulan Data

Data adalah fakta, angka, atau informasi lain yang dapat digunakan untuk menganalisis atau mengambil keputusan. Data yang dipilih untuk pengujian akan mempengaruhi tingkat akurasi [7]. Data dapat berupa angka, teks, gambar, atau kombinasi dari jenis-jenis tersebut. Data-data sedang dieksplorasi untuk aplikasi intensif pemrosesan di kamera seluler, IoT, dan mobil self-driving berkat model pengenalan video yang ditingkatkan [8]. Penelitian ini, data yang digunakan berupa video yang diperoleh dari Smartphone. Beberapa gambar data tersebut ditampilkan di gambar 2.



Gambar 2. Data Wajah

Setelah video (source) di input, selanjutnya data tersebut diolah untuk diproses dan digunakan untuk mendeteksi landmark mata. Proses ini dilakukan dengan menggunakan algoritma komputer yang dapat mengenali ciri-ciri mata dan menandai titik-titik penting pada mata tersebut. Hasil dari proses ini akan digunakan untuk aplikasi seperti pengenalan wajah, pengendalian perangkat melalui gerakan mata atau analisis ekspresi. Analisis ekspresi merupakan suatu sistem yang digunakan untuk mengidentifikasi ekspresi apa yang ditampilkan oleh seseorang [9]. Aplikasi ini adalah teknologi yang sangat berguna dan banyak digunakan dalam berbagai bidang, seperti keamanan, medis, dan hiburan. Flowchart pengolahan data metode yang diusulkan di tampilan di gambar 3.



Gambar 3. Flowchart preprocessing data

Untuk mengambil video sebagai masukan dalam program Python, kami menggunakan fungsi cv2.VideoCapture dari pustaka OpenC, Sebuah pustaka

yang banyak digunakan untuk pemrosesan gambar [10]. Fungsi ini memungkinkan untuk membuka file video atau merekam video dari kamera, dan membaca bingkai video satu per satu. Berikut adalah contoh bagaimana Anda dapat menggunakan fungsi cv2.VideoCapture untuk merekam video dari kamera dan menampilkan bingkai di jendela.

```
cap = cv2.VideoCapture
```

Gambar 4. Perintah python untuk menginput data uji

Frame adalah variabel yang mewakili gambar. Setiap frame memiliki nilai warna yang identik di setiap piksel. Ini diterapkan ke piksel lain dalam setiap frame, dan setiap frame dibandingkan dalam hal waktu [11]. Atribut bentuk gambar dalam format array NumPy adalah tuple yang berisi tinggi, lebar, dan jumlah saluran gambar. Variabel _ adalah placeholder yang digunakan untuk mengabaikan nilai yang tidak diperlukan. Dalam Script sistem, nilai image_height, image_width, dan _ ditetapkan ke nilai tupel bentuk yang sesuai. Misalnya, jika frame adalah gambar 480 x 640 dengan 3 saluran warna (merah, hijau, dan biru), maka frame.shape akan menjadi (480, 640, 3), dan variabelnya akan diberi nilai sesuai dengan yang diinput. Perintah untuk mengatur ukuran frame ditampilkan di gambar 5.

```
image_height = 480  
image_width = 640  
_ = 3
```

Gambar 5. Mengatur lebar dan panjang frame video

Atribut flags dari array NumPy adalah kamus yang berisi informasi tentang tata letak memori dari array. Salah satu kunci dalam kamus ini adalah dapat ditulisi, yaitu nilai boolean yang menentukan apakah data dalam larik dapat ditulisi atau tidak. Dalam sript sistem, flag yang dapat diatur ke False, yang berarti bahwa data dalam larik tidak dapat diubah. Ini berguna untuk mencegah modifikasi yang tidak disengaja pada larik, atau untuk memastikan bahwa data asli tidak diubah. Berikut adalah contoh bagaimana menggunakan atribut flags dan fungsi salin. Potongan script ditampilkan digambar 6.

```
# Create an array with some data  
data = np.array([1, 2, 3, 4, 5])  
  
# Make a copy of the array that is not writable  
data_copy = data.copy()  
data_copy.flags.writable = False  
  
# Try to modify the original array  
data[0] = 10  
  
# Try to modify the copy  
data_copy[1] = 20  
  
# Print the arrays to see the results  
print(data) # prints [10, 2, 3, 4, 5]  
print(data_copy) # prints [1, 2, 3, 4, 5]
```

Gambar 6. Penggunaan atribut flag

Frame adalah variabel yang merepresentasikan gambar dalam format array NumPy. Fungsi `cv2.cvtColor` adalah bagian dari pustaka OpenCV dan digunakan untuk mengkonversi gambar dari satu ruang warna ke ruang warna lainnya. Argumen pertama `cv2.cvtColor` adalah gambar yang akan dikonversi, dan argumen kedua adalah flag yang menentukan konversi yang akan dilakukan. Dalam hal ini, flag `cv2.COLOR_BGR2RGB` digunakan, yang menunjukkan bahwa gambar harus dikonversi dari ruang warna BGR ke ruang warna RGB. Ruang warna BGR adalah ruang warna yang digunakan oleh OpenCV untuk merepresentasikan gambar, dimana setiap piksel diwakili oleh tiga saluran 8-bit (biru, hijau, dan merah). Ruang warna RGB adalah ruang warna yang lebih umum digunakan untuk merepresentasikan gambar, di mana setiap piksel juga diwakili oleh tiga saluran 8-bit (merah, hijau, dan biru). Fungsi `cv2.cvtColor` mengembalikan gambar yang dikonversi sebagai larik NumPy, yang kemudian ditetapkan ke variabel gambar. Berikut adalah contoh bagaimana kami menggunakan fungsi `cv2.cvtColor` untuk mengonversi gambar dari BGR ke RGB yang ditampilkan pada gambar 7.

```
# Convert the image to RGB format
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

Gambar 7. BRG ke RGB

Setelah video diubah menjadi RGB, langkah berikutnya adalah mencari ukuran framenya. Ukuran framenya kemudian diubah menjadi 480 x 640. Untuk melihat contoh implementasinya, Gambar 8 menampilkan potongan script Python yang digunakan dalam pengolahan data input untuk sistem yang diusulkan.

```
cap = cv2.VideoCapture("datauji/frequency/26.mp4")
while True:
    _, frame = cap.read()
    image_height, image_width, _ = frame.shape
    frame.flags.writeable = False
    image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
```

Gambar 8. Proses data di python

2.2. Face Detection

Face detection atau deteksi wajah merupakan proses pengenalan atau identifikasi wajah manusia dalam gambar atau video. Proses ini dapat digunakan dalam berbagai aplikasi, seperti pengamanan kamera, pengenalan pelanggan di toko, atau pengenalan wajah dalam aplikasi ponsel. Deteksi wajah merupakan salah satu bagian dari computer vision. Teknologi computer vision juga menggunakan komputer untuk memperoleh informasi tingkat tinggi dari gambar berbentuk digital [12]. Teknologi yang digunakan dalam deteksi wajah mencakup analisis citra, pembelajaran mesin, dan algoritma pengenalan wajah. Deteksi wajah ini bahkan sudah diimplementasikan ke mobil otomatis. Mobil otomatis atau mobil tanpa pengemudi adalah mobil yang dapat menganalisis atau mengenali kondisi yang ada di jalan raya dengan menempatkan rangkaian sensor yang

tersebar di seluruh badan mobil. Gambar 9 menampilkan contoh deteksi wajah dengan bantuan komputer [13].



Gambar 9. Deteksi Wajah

2.3. Mediapipe Facemesh Landmark

MediaPipe adalah framework open-source yang dikembangkan oleh Google untuk membangun dan menerapkan pipeline machine learning multimodal lintas platform. Ini menyediakan kumpulan komponen pra-bangun yang dapat digunakan kembali untuk tugas-tugas seperti deteksi objek, deteksi landmark wajah, dan pelacakan tangan. MediaPipe dirancang untuk menyederhanakan pengembangan dan penerapan model pembelajaran mesin dengan menyediakan antarmuka yang konsisten di berbagai platform dan perangkat. Ini juga menyediakan seperangkat alat untuk memvisualisasikan dan men-debug pipeline, membuatnya mudah untuk memahami dan memecahkan masalah sistem. MediaPipe mendukung berbagai bahasa dan platform, seperti android, iOS, dan web, serta C++, Java, dan Python. Penelitian ini mengimplementasikan library mediapipe dengan Bahasa pemrograman python versi 3.10.6. Gambar 10 menampilkan pemanggilan library mediapipe dipython.

```
import itertools
import cv2
import mediapipe as mp
import numpy as np
import cvzone
from cvzone.PlotModule import LivePlot
```

Gambar 10. Mediapipe python

Setelah library Mediapipe terinstal, langkah selanjutnya adalah memanggil fungsi mediapipe untuk mendeteksi wajah. Fungsi ini merupakan bagian penting dalam proses deteksi wajah, karena mediapipe menyediakan berbagai komponen dan metode yang dapat digunakan secara efisien dan akurat. Dalam potongan script yang ditampilkan pada Gambar 11, sistem yang dibuat menggunakan fungsi tersebut untuk memanggil komponen deteksi wajah dari library mediapipe. Hal ini memungkinkan untuk mengakses fitur-fitur deteksi

wajah dan menerapkannya pada data input yang diberikan. Dengan menggunakan fungsi mediapipe, penelitian dapat memperoleh hasil deteksi wajah yang lebih akurat dan efisien.

```
# mediapipe utils
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
mp_face_mesh = mp.solutions.face_mesh
face_mesh = mp_face_mesh.FaceMesh(
    max_num_faces=1,
    refine_landmarks=True,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5)
```

Gambar 11. Fungsi deteksi wajah mediapipe

Beberapa fitur di library mediapipe adalah facemesh landmark. FaceMesh Landmark adalah teknologi yang dikembangkan oleh Google yang menggunakan pembelajaran mesin untuk mendeteksi dan menemukan titik-titik penting pada wajah seseorang, mirip dengan landmark wajah. FaceMesh Landmark menggunakan model mesh 3D untuk merepresentasikan wajah, dan mendeteksi 68 titik kunci yang sesuai dengan berbagai bagian wajah, seperti mata, hidung, mulut, dan garis rahang. Poin kunci ini dapat digunakan untuk berbagai aplikasi seperti animasi wajah, virtual dan augmented reality, atau bahkan untuk pengenalan wajah. Teknologi ini dirancang agar sangat akurat dan efisien, bahkan saat berhadapan dengan wajah yang sebagian tertutup atau bergerak. Google menyediakan library MediaPipe Facemesh sebagai solusi open-source untuk mendeteksi landmark wajah di ponsel, web, dan desktop. Gambar 12 menampilkan fungsi facemesh landmark dengan Mediapipe python.



Gambar 12. Facemesh landmark

2.4. Eyes Detection

Eyes detection atau deteksi mata adalah proses mengidentifikasi atau menemukan mata dalam gambar atau video. Ini dapat digunakan dalam berbagai aplikasi seperti pengenalan wajah, pelacakan tatapan, atau interaksi manusia-komputer. Teknologi deteksi mata biasanya menggunakan analisis gambar, pembelajaran mesin, dan algoritme visi komputer untuk mendeteksi dan menemukan mata dalam sebuah gambar. Beberapa fitur utama yang digunakan untuk mendeteksi mata antara lain bentuk mata, posisi relatif mata terhadap fitur

wajah lainnya, serta tekstur dan warna mata. Ada berbagai algoritma dan teknik yang dikembangkan untuk deteksi mata, seperti Haar Cascades, algoritma Viola-Jones, HOG (Histograms of Oriented Gradients), Yolo (You Only Look Once), dan CNN (Convolutional Neural Network). Penelitian oleh [14] menunjukkan bahwa penggunaan algoritma yolo untuk mendeteksi objek memperoleh akurasi yang baik. Terdapat beberapa pustaka dan kerangka kerja yang menyediakan fungsionalitas deteksi mata adalah OpenCV, dlib, dan MediaPipe. Gambar mata manusia pada umumnya ditampilkan di gambar 13.



Gambar 13. Mata manusia

3. Hasil dan Pembahasan

Pengujian adalah proses untuk mengevaluasi kualitas atau kinerja suatu produk atau sistem. Ini dapat dilakukan dengan menggunakan berbagai metode, seperti uji coba, analisis, atau inspeksi, untuk menentukan apakah produk atau sistem tersebut memenuhi spesifikasi yang ditentukan atau tidak. Tujuan dari pengujian adalah untuk menemukan dan mengoreksi kesalahan atau masalah sebelum produk atau sistem. Pengujian di penelitian dilakukan untuk mengetahui akurasi sistem untuk mendeteksi mata manusia. Pengujian ini menggunakan 56 video yang berdurasi 5-10 detik dan diambil dengan pencahayaan yang baik (terang). Objek video terdiri dari pria dan wanita yang berusia antara 20-22 tahun. Objek untuk data testing melakukan dua kondisi, yaitu mata terbuka dan mata tertutup. Gambar 14 menampilkan contoh data pengujian yang digunakan di penelitian ini.



Gambar 14. Video Data Testing

Untuk menghitung akurasi kami menggunakan metode confusion matriks. Confusion matriks adalah pengukuran yang sangat populer digunakan untuk memecahkan masalah klasifikasi, dan dapat diterapkan pada klasifikasi biner serta untuk masalah klasifikasi

multi kelas [15]. Selain memecahkan, confusion matrik biasa digunakan untuk meringkas pekerjaan yang berkaitan klasifikasi yang merupakan masalah yang sangat penting di banyak bidang seperti matematika terapan, geostatistik, penambangan data, analitik teks, keuangan, biomedis, bioteknologi [16] [17]. Perhitungan akurasi dengan metode confusion matriks ditampilkan di persamaan 1.

$$\text{Accuracy} = \left(\frac{TP+TN}{TP+FP+FN+TN} \right) \times 100\% \quad (1)$$

Parameter TP, FP, TN, dan FN memiliki peran penting dalam evaluasi kinerja suatu sistem atau model. TP (true positive) digunakan untuk menggambarkan jumlah data positif yang secara benar diprediksi sebagai positif oleh model. FP (false positive) mengindikasikan jumlah data negatif yang keliru diprediksi sebagai positif. TN (true negative) mewakili jumlah data negatif yang benar-benar diprediksi sebagai negatif. Sedangkan FN (false negative) mencerminkan jumlah data positif yang salah diprediksi sebagai negatif. Informasi lebih rinci mengenai confusion matriks untuk penelitian ini dapat ditemukan di tabel 1.

Tabel 1. Keterangan

Huruf	Keterangan
P	Kelas prediksi
A	Kelas aktual
M	Mata
N	Bukan Mata
TP	Diprediksi positif dan itu benar
TN	Diprediksi negatif dan itu benar
FN	Diprediksi negatif dan itu salah
FP	Diprediksi positif dan itu salah

Tabel confusion matrix adalah representasi visual yang berguna dalam mengevaluasi kinerja sistem deteksi mata. Tabel confusion matriks menjelaskan bahwa baris mewakili hasil kelas actual dan kolom mewakili hasil kelas prediksi [19]. Confusion matrix ini memberikan gambaran tentang seberapa baik sistem mampu mengklasifikasikan data dalam kategori yang benar. Informasi lebih lanjut mengenai confusion matrix dan interpretasinya dapat ditemukan di tabel 2.

Tabel 2. Confusion Matriks Sistem Deteksi Mata

	P	M	N
A			
M		56(TP)	0(FN)
N		0(FP)	0(TN)

$$\text{Akurasi} = \left(\frac{56+0}{56+0+0+0} \right) \times 100\% \quad (2)$$

$$\text{Akurasi} = \left(\frac{56}{56} \right) \times 100\% \quad (3)$$

Akurasi= 100 %.

Selain itu, dalam penelitian ini kami melakukan perbandingan antara metode yang kami usulkan dengan metode yang telah ada sebelumnya dalam deteksi mata. Dengan membandingkan hasil ini, kami dapat memperoleh pemahaman yang lebih baik tentang keunggulan dan kelemahan masing-masing metode dalam konteks deteksi mata. Informasi lebih detail tentang hasil perbandingan ini dapat ditemukan dalam Tabel 3.

Tabel 3. Perbandingan metode yang diusulkan dengan metode lainnya

Metode	Akurasi
Eyes aspect ratio[3]	96.85%
Measurement vertical and horizontal at part of eyes [20]	90%
Facelandmark and u-LBPH [21]	95.5%
SNEO and VME [22]	95.04%
Mediapipe for eyes detection	100%

Seperti yang telah disampaikan di bagian pendahuluan bahwa metode yang kami usulkan dapat membedakan antara kondisi mata terbuka dan tertutup. Bila mata tertutup, maka titik-titik disekitar mata akan berwarna hijau. Sedangkan bila mata terbuka, titik-tik tersebut akan berwarna kuning. Kondisi tersebut merupakan kondisi fisiologis normal dari mata. Kondisi mata terbuka menunjukkan bahwa mata sedang dalam keadaan normal dan siap untuk melihat. Sedangkan kondisi mata tertutup menunjukkan bahwa mata sedang dalam keadaan istirahat atau tidur. Kedua kondisi ini sangat penting untuk kesehatan mata dan dapat mempengaruhi kemampuan visual seseorang. Gambar 15 menampilkan potongan script python untuk perintah tersebut.

```

if distance1*100/np.mean(eyes_areas) < 20:
    INITIAL_COLOR = (0,255,0)
    B_COLOR = (0,255,0)
    MEMBER_OF_BLINKS += 1
elif counter != 0:
    counter += 1
    if counter > 3:
        counter = 0
    pass
elif distance1*100/np.mean(eyes_areas) < 50:
    pass
else:
    INITIAL_COLOR = (0,0,255)
    B_COLOR = (0,255,255)
    eyes_areas.append(distance1)
    
```

Gambar 15. Potongan Script

Script di atas memiliki fungsi untuk melakukan perubahan warna. Warna pada gambar akan berubah sesuai dengan kondisi mata yang terdeteksi, yaitu mata terbuka atau mata tertutup. Untuk memberikan gambaran yang lebih jelas tentang perbedaan antara kedua kondisi tersebut, kami menampilkan Gambar 16. Gambar ini mengilustrasikan dengan visual perbedaan karakteristik antara mata terbuka dan mata tertutup.



Gambar 16. Landmark mata

4. Kesimpulan

Dalam penelitian ini, telah dikembangkan metode baru untuk mendeteksi mata manusia menggunakan library Mediapipe. Teknologi deteksi mata memiliki peran yang sangat penting dalam berbagai bidang dan aplikasi, seperti sistem pengenalan wajah, analisis ekspresi wajah, pengendalian perangkat, dunia medis, dan sistem keamanan. Metode yang diusulkan dalam penelitian ini memperoleh akurasi 100% dalam mendeteksi mata, menunjukkan kemajuan yang signifikan dalam bidang pelacakan mata. Metode ini dapat diaplikasikan pada berbagai bidang seperti psikologi, kedokteran, dan interaksi manusia-komputer.

Penelitian ini menggunakan library mediapipe dalam bahasa pemrograman Python versi 3.10.6. Metode yang dikembangkan menawarkan solusi yang lebih efisien dan hemat biaya dalam mendeteksi mata dibandingkan dengan metode sebelumnya. Selain itu, penelitian ini menunjukkan potensi efektifitas penggunaan perpustakaan Mediapipe Python untuk penelitian dan aplikasi pelacakan mata.

Namun, terdapat beberapa keterbatasan dalam penelitian ini yang perlu dipertimbangkan. Pertama, ukuran sampel yang digunakan relatif kecil, sehingga penelitian di masa depan dapat memperluas sampel untuk meningkatkan generalisasi temuan. Kedua, penelitian ini hanya menguji metode pada satu populasi, sehingga penting untuk mengeksplorasi penggunaan metode pada populasi yang berbeda, seperti anak-anak atau orang dengan kondisi medis tertentu. Selanjutnya, penelitian di masa depan dapat melihat penggunaan metode ini dalam tugas pelacakan mata lainnya, seperti pelebaran pupil atau gerakan saccadic.

Secara keseluruhan, hasil penelitian ini menunjukkan potensi yang baik dalam penggunaan perpustakaan Mediapipe Python untuk penelitian dan aplikasi pelacakan mata. Penelitian di masa depan dapat memperluas temuan yang ada dan mengeksplorasi penggunaan metode ini pada populasi yang berbeda dengan tujuan yang sama.

Daftar Pustaka

- [1] A. A. M. Suradi, "Deteksi Jalan Berdasarkan Segmentasi Warna HSV Dalam Penerapan Mobil Otonom (Autonomous Car)," 2021.
- [2] M. Knapik and B. Cyganek, "Fast eyes detection in thermal images," *Multimed Tools Appl*, vol. 80, no. 3, pp. 3601–3621, Jan. 2021, doi: 10.1007/s11042-020-09403-6.
- [3] C. Dewi, R.-C. Chen, X. Jiang, and H. Yu, "Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks," *PeerJ Comput. Sci.*, vol. 8, p. e943, Apr. 2022, doi: 10.7717/peerj-cs.943.
- [4] F. Zhang *et al.*, "MediaPipe Hands: On-device Real-time Hand Tracking." arXiv, Jun. 17, 2020. Accessed: Feb. 01, 2023. [Online]. Available: <http://arxiv.org/abs/2006.10214>
- [5] C. Lugaresi *et al.*, "MediaPipe: A Framework for Building Perception Pipelines." arXiv, Jun. 14, 2019. doi: 10.48550/arXiv.1906.08172.
- [6] G. Güney *et al.*, "Video-Based Hand Movement Analysis of Parkinson Patients before and after Medication Using High-Frame-Rate Videos and MediaPipe," *Sensors*, vol. 22, no. 20, Art. no. 20, Jan. 2022, doi: 10.3390/s22207992.
- [7] "Automatic Detection of Tomato Leaf Deficiency using Soft Computing Technique," *IJEAT*, vol. 9, no. 2, pp. 5406–5410, Dec. 2019, doi: 10.35940/ijeat.A1045.129219.
- [8] D. Kondratyuk *et al.*, "MoViNets: Mobile Video Networks for Efficient Video Recognition," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 16015–16025. doi: 10.1109/CVPR46437.2021.01576.
- [9] M. F. Rasyid, "Comparison Of LBPH, Fisherface, and PCA For Facial Expression Recognition of Kindergarten Student," *International Journal Education and Computer Studies (IJECS)*, vol. 2, no. 1, Art. no. 1, May 2022, doi: 10.35870/ijeecs.v2i1.625.
- [10] A. H. Ahamad, N. Zaini, and M. F. A. Latip, "Person Detection for Social Distancing and Safety Violation Alert based on Segmented ROI," in *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Aug. 2020, pp. 113–118. doi: 10.1109/ICCSCE50387.2020.9204934.
- [11] A. P. Ismail, F. A. A. Aziz, N. M. Kasim, and K. Daud, "Hand gesture recognition on python and opencv," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1045, no. 1, p. 012043, Feb. 2021, doi: 10.1088/1757-899X/1045/1/012043.
- [12] A. A. M. Suradi, M. F. Rasyid, and N. Nasaruddin, "Sistem Perhitungan Jumlah Kendaraan Berbasis Computer Vision," *SISITI : Seminar Ilmiah Sistem Informasi dan Teknologi Informasi*, vol. 11, no. 1, Art. no. 1, Aug. 2022, Accessed: Jan. 26, 2023. [Online]. Available: <http://ejournal.diponegara.ac.id/index.php/sisiti/article/view/950>
- [13] A. A. Mahersatillah, Z. Zainuddin, and Y. Yusran, "Unstructured Road Detection and Steering Assist Based on HSV Color Space Segmentation for Autonomous Car," in *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Dec. 2020, pp. 688–693. doi: 10.1109/ISRITI51436.2020.9315452.
- [14] F. Rasyid, Mar'atuttahirah, M. S. Mustafa, and A. A. M. Suardi, "Menghitung Akurasi Sistem

- Deteksi Masker Berdasarkan Arah Pandangan Kepala Objek,” *Computer Science Research and Its Development Journal*, vol. 14, no. 3, Art. no. 3, Dec. 2022, doi: 10.22303/csrid.14.3.2022.203-214.
- [15] “Confusion Matrix - an overview | ScienceDirect Topics.”
<https://www.sciencedirect.com/topics/engineering/confusion-matrix> (accessed Jul. 04, 2022).
- [16] M. Rasyid, Z. Zainuddin, and A. Andani, “Early Detection of Health Kindergarten Student at School Using Image Processing Technology,” in *Proceedings of the 1st International Conference on Science and Technology, ICOST 2019, 2-3 May, Makassar, Indonesia*, Makassar, Indonesia: EAI, 2019. doi: 10.4108/eai.2-5-2019.2284609.
- [17] “Confusion Matrix - an overview | ScienceDirect Topics.” Accessed: Jul. 04, 2022. [Online]. Available:
<https://www.sciencedirect.com/topics/engineering/confusion-matrix>
- [18] Moh. A. Hasan, Y. Riyanto, and D. Riana, “Grape leaf image disease classification using CNN-VGG16 model,” *J. Teknol. dan Sist. Komput*, vol. 9, no. 4, pp. 218–223, Oct. 2021, doi: 10.14710/jtsiskom.2021.14013.
- [19] J. Xu, Y. Zhang, and D. Miao, “Three-way confusion matrix for classification: A measure driven view,” *Information Sciences*, vol. 507, pp. 772–794, Jan. 2020, doi: 10.1016/j.ins.2019.06.064.
- [20] F. Utaminingrum, A. D. Purwanto, M. R. R. Masruri, K. Ogata, and I. K. Somawirata, “Eye Movement and Blink Detection for Selecting Menu On-Screen Display Using Probability Analysis Based on Facial Landmark.” *ICIC International 学会*, 2021. doi: 10.24507/ijicic.17.04.1287.
- [21] D. A. Navastara, W. Y. M. Putra, and C. Fatichah, “Drowsiness Detection Based on Facial Landmark and Uniform Local Binary Pattern,” *J. Phys.: Conf. Ser.*, vol. 1529, no. 5, p. 052015, May 2020, doi: 10.1088/1742-6596/1529/5/052015.
- [22] M. Wang *et al.*, “Multidimensional Feature Optimization Based Eye Blink Detection Under Epileptiform Discharges,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 905–914, 2022, doi: 10.1109/TNSRE.2022.3164126.