

PENERAPAN *HAND GESTURE RECOGNITION* SEBAGAI MEDIA KONTROL PRESENTASI APLIKASI *POWERPOINT*

Dody Khairianto, Rahmad Firdaus

Teknik Informatika, Universitas Muhammadiyah Riau
Jalan KH. Ahmad Dahlan No.88 Pekanbaru, Indonesia
200401107@student.umri.ac.id

ABSTRAK

Presentasi merupakan alat komunikasi yang berperan penting dalam menyampaikan informasi dan ide kepada audiens. Salah satu media yang digunakan adalah aplikasi *PowerPoint*. Aplikasi ini memungkinkan pengguna untuk memberikan konten presentasi dengan elemen visual, seperti teks, gambar, dan animasi. Aplikasi ini termasuk dalam salah satu program komputer yang memerlukan suatu perangkat kendali seperti *keyboard* dan *mouse*. Akan tetapi penggunaan perangkat-perangkat tersebut terkadang dapat mengganggu pengguna dalam melakukan kontrol terhadap aplikasi ini. Oleh karena itu penulis merancang sebuah *prototype* untuk mempermudah kontrol aplikasi ini. Solusinya adalah menggunakan kamera sebagai perangkat input *real-time* menggunakan bahasa pemrograman *Python* dengan *library OpenCV* untuk melakukan pemrosesan video, *MediaPipe* yang berperan sebagai *framework* untuk mendapatkan *landmark* tangan, dan *CVZone* untuk mempermudah melakukan klasifikasi gestur tangan. Hasil dari perancangan dan pengujian yang telah dilakukan menunjukkan bahwa untuk berpindah ke *slide* selanjutnya memerlukan waktu responsif rata-rata 2.075 detik, berpindah ke *slide* sebelumnya mendapatkan rata-rata 1.7265 detik, masuk mode layar penuh mendapatkan waktu rata-rata 2.0305 detik, dan keluar mode layar penuh mendapatkan waktu rata-rata 1.8665 detik. Pengujian pada jarak 1 meter mendapatkan rata-rata dalam waktu 1,766 detik. Sedangkan pada jarak 2 meter program mampu mengenali gestur tangan dalam waktu 2,08325 detik.

Kata kunci: *Hand Gesture Recognition, Presentasi, PowerPoint*

1. PENDAHULUAN

Presentasi merupakan alat komunikasi untuk menyampaikan informasi dan ide kepada audiens [1]. Presentasi harus disampaikan dengan jelas dan terperinci. *PowerPoint* merupakan aplikasi presentasi multimedia yang memungkinkan pengguna untuk menyajikan informasi melalui gabungan teks, audio, dan elemen visual secara bersamaan [2]. Aplikasi ini termasuk dalam salah satu program komputer yang memerlukan suatu perangkat kendali dalam proses pengoperasiannya. Umumnya, pemilihan perangkat ini perlu memperhatikan faktor psikologis dan kenyamanan pengguna dalam interaksi alami dengan teknologi [10]. Untuk menggeser *slide* presentasi aplikasi ini, biasanya pengguna menggunakan *mouse* atau *keyboard*. Akan tetapi terkadang penggunaan perangkat *Human-Computer Interaction* (HCI) dapat mengganggu pengguna dalam melakukan kontrol terhadap aplikasi ini [3]. Untuk itu dibutuhkan sebuah sistem kendali yang dapat mempermudah melakukan kontrol terhadap aplikasi ini.

Hardware komputer terus berkembang pesat setiap tahunnya. Hal ini dapat dilihat dari berbagai perangkat keras yang telah diluncurkan, seperti *processor*, *motherboard*, layar LCD, LED, sistem penyimpanan seperti *hddisk* dan *flashdisk* yang memiliki kapasitas besar, proyektor, dan teknologi *bluetooth* [4]. Kamera juga menjadi salah satu perangkat keras yang diluncurkan pada perkembangan ini. Kamera merupakan perangkat untuk merekam citra suatu objek pada permukaan yang responsif terhadap cahaya [5].

Kemajuan teknologi informasi dan media terkini telah mendorong penelitian dalam bidang sistem HCI [6]. Terutama studi dalam interaksi manusia dan komputer yang telah memperkenalkan pendekatan menarik, termasuk teknik antarmuka berbasis informasi gerak dari gambar video tanpa peralatan mahal. Pendekatan ini didasarkan pada *computer vision* dan bertujuan untuk mengembangkan sistem antarmuka komputer-manusia yang efektif [4]. *Hand Gesture Recognition* (HGR) merupakan sistem yang mampu mendeteksi gestur tangan dalam video secara *real-time*, dengan gestur tangan yang diklasifikasikan untuk kepentingan tertentu. Mengenali gerakan tangan merupakan salah satu masalah utama yang penting dalam bidang *computer vision* [6]. HGR berbasis kerangka sangat populer karena kemampuannya terhadap mengenali gestur tangan dengan berbagai variasi latar belakang dan cahaya [7].

Rahmawati, dkk telah mengembangkan *pointing device* tanpa perangkat keras seperti *mouse*. Penelitian tersebut menggunakan model *hand landmark*, di mana jari tangan akan diidentifikasi dengan menjadi 21 simbol titik. *Pointing device* ini memungkinkan kursor dapat bergerak sesuai perintah tanpa perlu sentuhan tangan, menggunakan kamera laptop atau komputer sebagai pengganti *mouse* [8].

Indriani, dkk telah melakukan pengembangan aplikasi panduan pengguna sederhana menggunakan *framework MediaPipe*. Panduan ini berupa panduan yang umumnya dikenal sebagai dokumentasi tentang komunikasi teknis atau manual dalam suatu sistem tertentu untuk membantu pengguna. Aplikasi ini

menangkap gambar real-time menggunakan *Kinect*, kemudian melatih berbagai data gerakan tangan, mengidentifikasi setiap gerakan tangan, dan mengenali gerakan tangan untuk menyampaikan informasi berdasarkan gerakan tangan di aplikasi panduan pengguna sistem. Pengguna dapat mengarsipkan informasi panduan pengguna berdasarkan gerakan tangan yang telah dikenali. [9].

Peral, dkk juga telah mengembangkan pendekatan *deep learning* yang efisien dan andal yang memungkinkan pengguna berinteraksi dengan robot menggunakan gestur tangan. Pendekatan ini terdiri dari dua modul yang bekerja menggunakan *landmark* tangan 2D. Metode yang digunakan telah mencapai akurasi 87,5% dan berjalan pada 10 frame per detik. Terakhir, peneliti melakukan eksperimen dengan robot IVO peneliti untuk memvalidasi kerangka kerja selama proses interaksi [10].

Dari pembahasan yang telah dilakukan, penulis bertujuan untuk merancang sebuah *prototype* yang memanfaatkan *hand gesture recognition* untuk melakukan kontrol aplikasi *PowerPoint* menggunakan bahasa pemrograman *Python* dengan *library OpenCV*, *MediaPipe* dan *CVZone*. Hal ini bertujuan untuk mempermudah pengguna dengan meningkatkan mobilitas dalam melakukan kontrol terhadap aplikasi ini. *Prototype* ini beroperasi secara *real-time* menggunakan kamera dalam melakukan pengenalan gesur tangan. Harapan dari implementasi rancangan ini adalah memberikan kemudahan kepada pengguna dalam melakukan kontrol terhadap aplikasi *PowerPoint*.

2. TINJAUAN PUSTAKA

2.1. OpenCV

Open Source Computer Vision Library (*OpenCV*) adalah *Application Peripheral Interface* (API) yang dikembangkan oleh *Intel* yang dapat digunakan untuk banyak aplikasi pengolah gambar dan *computer vision*. *OpenCV* bertujuan menyediakan alat dasar yang diperlukan untuk memecahkan masalah *computer vision*. Aplikasi *computer vision* seringkali membutuhkan *Machine Learning* (ML) atau metode *Artificial Intelligence* (AI) lainnya yang beberapa di antaranya tersedia dalam *OpenCV* [11].

2.2. CVZone

CVZone merupakan sebuah *package* berbasis *Python* yang dirancang untuk membuat pengolahan gambar dan implementasi fungsi kecerdasan buatan menjadi lebih sederhana. *Package* ini dapat membuat integrasi algoritma pengolahan gambar dan kecerdasan buatan ke dalam proyek *python* menjadi lebih mudah. *Package* ini dirancang dengan memanfaatkan dua *library*, yaitu *OpenCV* dan *Mediapipe* [12].

2.3. MediaPipe

MediaPipe merupakan sebuah *famework* yang dirancang khusus untuk mendukung pengembangan

dalam bidang *machine learning*. Platform ini memiliki kemampuan untuk mengolah data *time-series*, seperti video dan audio, dan sebagainya [13]. Algoritma yang digunakan dalam *MediaPipe* sendiri adalah *Convolutional Neural Networks* (CNN) [14]. *Framework* ini memungkinkan kita untuk mendapatkan kerangka gestur tangan.

2.4. PowerPoint

PowerPoint adalah sebuah perangkat lunak presentasi multimedia yang memberikan kemampuan kepada pengguna untuk menyampaikan informasi dengan menggunakan kombinasi teks, audio, dan elemen visual secara bersamaan. Dengan menggunakan *PowerPoint*, pengguna dapat membuat presentasi yang dinamis dan menarik dengan menggabungkan berbagai unsur multimedia, seperti teks yang informatif, audio yang mendukung, dan elemen visual yang menarik perhatian audiens [2].

2.5. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan suatu bagian dalam domain *deep learning* yang berfungsi dengan cara mendeteksi dan mengelompokkan objek pada gambar secara otomatis. Proses ini terjadi melalui penerimaan input data berukuran $m \times n$. CNN sendiri memiliki keunggulan dalam akurasi klasifikasi gambar yang tinggi, menjadikannya alat yang efektif dalam tugas-tugas pengenalan gambar [15].

2.6. Computer Vision

Computer vision adalah pemrosesan tingkat tinggi yang melibatkan pemahaman atau sesuatu yang “masuk akal” dari sekelompok objek yang dikenali, sebagai analisis citra dan akhirnya melakukan fungsi yang mirip dengan penglihatan manusia. *Computer vision* tingkat tinggi mencoba meniru kognisi manusia dan kemampuan untuk mengambil keputusan sesuai dengan informasi yang terdapat dalam citra [11].

2.7. PyCharm

Pycharm adalah perangkat lunak IDE (*integrated development environmental*) untuk sistem operasi windows yang mempermudah penggunaan dan pembuatan program dengan bahasa pemrograman *Python* [16]. Perangkat lunak ini merupakan IDE *Python* yang dibuat oleh *JetBrains*, yaitu perusahaan di belakang perangkat lunak *IntelliJ IDEA* yang merupakan IDE populer untuk *Java*. *PyCharm* adalah IDE yang bagus untuk mengerjakan proyek yang berisi banyak *script* yang berinteraksi satu sama lain. *PyCharm* dapat digunakan untuk setiap pengembang yang ingin membuat aplikasi perangkat lunak dengan *Python*, baik itu aplikasi web, aplikasi sains data, atau bahkan hanya skrip *Python* sederhana [17].

2.8. Python

Python merupakan suatu bahasa pemrograman yang berorientasi objek yang memungkinkan

pengguna untuk berinteraksi secara langsung. Dengan struktur data tingkat tinggi, bahasa ini terkenal sebagai bahasa pemrograman interpretatif yang menyediakan beragam fungsi. Bahasa pemrograman ini dibuat dengan fokus dalam kejelasan dan kemudahan dalam pemahaman kode, menjadikannya populer di kalangan pengembang. Dengan menggabungkan kemampuan dan kejelasan sintaks kode, *Python* diakui sebagai alat yang efektif dan efisien dalam pengembangan perangkat lunak [18].

2.9. Kamera

Kamera dapat dipahami sebagai sebuah perangkat yang berfungsi untuk merekam gambar melalui penggunaan lensa, yang kemudian mengalami proses pengolahan di dalam kamera sebelum akhirnya disimpan pada media penyimpanan yang tersedia. Lebih jauh lagi, kamera dapat dianggap sebagai suatu instrumen optik yang memiliki kemampuan untuk menangkap dan menyimpan gambar-gambar yang terletak di depannya. [19].

2.10. Hand Gesture

Isyarat tangan (*Hand Gesture*) merujuk pada gerakan atau posisi tangan yang digunakan sebagai salah satu bentuk komunikasi nonverbal yang dapat diimplementasikan dalam beragam aplikasi di berbagai bidang. Keberagaman penggunaannya mencakup komunikasi antara individu yang memiliki keterbatasan bicara atau pendengaran, pengendalian robot, interaksi antara manusia dan komputer (*Human-Computer Interaction/HCI*), implementasi otomatisasi dalam lingkungan rumah, serta penerapan dalam konteks aplikasi medis. [20].

2.11. Hand Gesture Recognition (HGR)

Pengenalan isyarat tangan (*hand gesture recognition*), merujuk pada bentuk interaksi antara manusia dan komputer atau mesin. Konsep dasar yang paling mendasar dalam pengenalan isyarat tangan adalah membuat interaksi yang alami, sesuai dengan kebiasaan manusia, antara pengguna dan sistem komputer. Tujuan dari pengenalan isyarat tangan ini adalah untuk memberikan kemampuan kepada pengguna untuk mengontrol perangkat atau menyampaikan informasi melalui gerakan tangan yang dikenali oleh sistem[21].

2.12. Hand Landmark

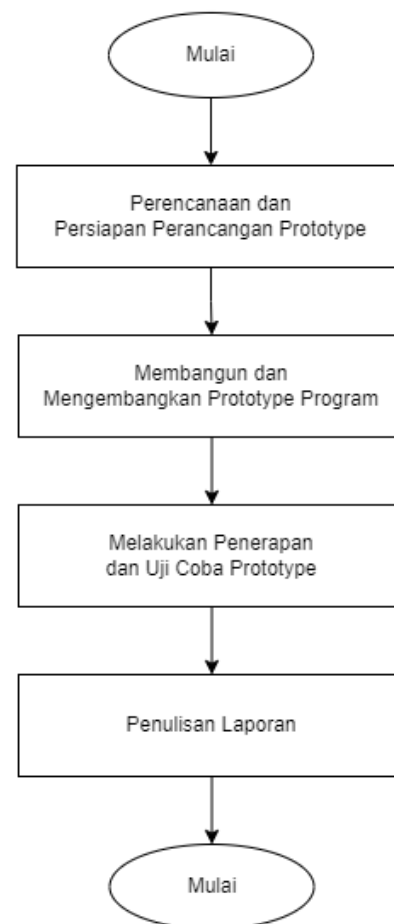
Hand landmark merupakan lokasi 21 titik pada tangan, yang ditentukan oleh koordinat yang berasal dari jari-jari yang terdeteksi pada citra tangan. Proses ini dilakukan melalui regresi, di mana model landmark tangan di *MediaPipe* menghasilkan prediksi koordinat secara langsung untuk setiap titik, menciptakan suatu representasi visual yang akurat dari struktur tangan pada citra yang terdeteksi [9].



Gambar 1. Hand Landmark di MediaPipe

3. METODE PENELITIAN

Bagian ini memuat uraian tentang langkah-langkah pelaksanaan proses pengembangan program, yaitu perencanaan alur kerja, implementasi dan pengujian rancangan *Prototype*, dengan memecahnya sehingga membuahkan alur terhadap penelitian tersebut:



Gambar 2. Tahapan penelitian

3.1. Perencanaan dan Persiapan Perancangan Prototype

Sebelum melakukan perancangan, perencanaan dan persiapan merupakan langkah yang penting dalam melakukan perancangan *prototype* yang efektif dan efisien. Proses ini dimulai dengan menggali informasi terhadap tujuan yang ingin dicapai melalui *prototype* ini. Penulis perlu melakukan perincian untuk setiap fitur dan fungsi yang akan diimplementasikan serta merancang struktur dasar *prototype*. Selain itu, perencanaan juga melibatkan pemilihan alat--alat yang akan digunakan untuk mendukung kebutuhan

rancangan ini. Dengan perencanaan dan persiapan yang baik, penulis dapat memastikan bahwa perancangan *prototype* dapat berjalan lancar dan menghasilkan solusi yang diharapkan.

3.2. Membangun dan Mengembangkan Prototype Program

Pada tahap ini, akan dijelaskan langkah-langkah alur kerja dari *prototype* yang telah direncanakan pada tahapan sebelumnya. Adapun beberapa langkah utama dalam proses ini terdiri dari merekam tangan menggunakan kamera, memproses dan mendapatkan gestur tangan, menentukan perintah yang akan dieksekusi, dan melakukan kontrol terhadap aplikasi *PowerPoint*.

3.2.1. Merekam Tangan Menggunakan Kamera

Untuk menjalankan program dan memperoleh informasi mengenai gestur tangan secara *real-time*, diperlukan kamera yang dapat melakukan penangkapan citra gambar secara berulang selama proses berlangsung. Fungsinya adalah untuk merekam pergerakan tangan secara terus-menerus, memungkinkan program untuk secara akurat mendapatkan dan merespons setiap gerakan yang dilakukan. Dengan adanya kamera, program dapat secara efisien mengumpulkan data visual yang diperlukan untuk mengidentifikasi gestur tangan dengan presisi, membuka peluang untuk pengembangan aplikasi yang lebih canggih dan responsif dalam mendukung interaksi manusia dengan teknologi secara lebih intuitif.

3.2.2. Memproses dan Mendapatkan Gestur Tangan

Proses pengambilan citra pada *prototype* ini akan dilakukan menggunakan *library OpenCV*, sebuah alat yang dirancang untuk mendeteksi dan mengolah gambar. Tujuan utama penggunaan *library* ini adalah untuk mendapatkan gambar tangan melalui kamera. Penggunaan *OpenCV* dalam proyek ini tidak hanya pada pengambilan gambar, tetapi juga memberikan dasar informasi yang nantinya diperlukan untuk analisis bentuk dan posisi tangan.

3.2.3. Menentukan Perintah Yang Akan Dieksekusi

Penentuan posisi *hand landmark* dapat dilakukan melalui penggunaan *library MediaPipe*, yang digunakan dalam *CVZone Package*. Informasi mengenai posisi *landmark* tangan ini kemudian digunakan sebagai titik acuan untuk mengeksekusi perintah yang akan dijalankan oleh sistem komputer. Dengan adanya metode ini, komputer dapat merespons pergerakan atau posisi tangan secara akurat, memungkinkan interaksi yang lebih efisien antara pengguna dan sistem. Proses ini memanfaatkan *library MediaPipe* dan *CVZone Package* untuk mengenali dan memetakan *landmark* tangan, yang kemudian menjadi

dasar acuan untuk menginterpretasikan perintah-perintah yang diberikan kepada komputer.



Gambar 3. Contoh Citra Landmark Tangan Kanan Yang Terdeteksi

3.2.4. Melakukan Kontrol Terhadap Aplikasi PowerPoint

Setelah *landmark* tangan berhasil didapatkan, langkah selanjutnya pada *prototype* ini adalah menjalankan serangkaian perintah yang memberikan kontrol terhadap aplikasi *PowerPoint*. Perintah-perintah ini dirancang untuk memungkinkan pengguna dalam melakukan kontrol aplikasi *PowerPoint*. Pengguna dapat dengan mudah beralih ke *slide* berikutnya, kembali ke *slide* sebelumnya, serta memasuki dan meninggalkan mode tampilan layar penuh.

3.3. Melakukan Penerapan dan Uji Coba Prototype

Prototype yang telah dibuat akan diterapkan dan diuji coba untuk melihat hasil dari *prototype* telah dirancang. Adapun metode dalam pengujian cobaan ini adalah dengan melakukan pengujian secara langsung di IDE *PyCharm* untuk melihat keberhasilan yang didapat oleh *prototype* ini dalam melakukan fungsinya. Berikutnya akan dilanjutkan ke uji coba waktu responsif program dengan menjalankannya dan menggunakannya langsung dengan aplikasi presentasi *PowerPoint*.

3.4. Penulisan Laporan

Penulisan laporan dilakukan setelah aplikasi diuji untuk menjabarkan hasil dari penelitian yang dilakukan dalam menyelesaikan permasalahan yang ada, yaitu:

- Bagaimana menerapkan *hand gesture recognition* sebagai input untuk meningkatkan mobilitas kontrol *slide* presentasi pada aplikasi *PowerPoint*?
- Bagaimana cara mengimplementasikan *library OpenCV*, *Mediapipe*, dan *CVZone* dalam penerapan *hand gesture recognition*?

4. HASIL DAN PEMBAHASAN

4.1. Membangun dan Mengembangkan Prototype Program

Sebelum mulai menulis kode *Python*, penulis perlu memulai dengan membuat proyek baru di *PyCharm*. Selanjutnya, melakukan instalasi beberapa *library Python* yang akan digunakan, seperti *OpenCV*, *MediaPipe*, *CVZone*, *PyAutoGui*, *Math*, *Numpy*, dan *Time*. *Library* ini diperlukan untuk membantu dalam perancangan *prototype* dalam hal fungsionalitas. Berikut ini adalah baris kode yang digunakan untuk mengimpor beberapa *library* tersebut ke dalam proyek:

```
import cv2
import numpy
import math
import time
import pyautogui
from cvzone.HandTrackingModule import HandDetector
```

4.2. Merekam Tangan Menggunakan Kamera

Diperlukan sebuah alat untuk melakukan penangkapan citra gambar secara terus menerus selama program dijalankan guna mendapatkan informasi gestur tangan. Oleh karena itu penulis membutuhkan perangkat kamera yang akan diintegrasikan ke sistem. Pada kasus ini penulis menggunakan kamera bawaan laptop. Untuk melakukan integrasi dari kamera ke program, penulis menggunakan *library OpenCV*. *Library* ini berperan sebagai API yang digunakan pada kode program *prototype* untuk mendapatkan gestur tangan secara *real-time*. Berikut merupakan kode *Python* yang digunakan:

```
width, height = 800, 600

cap = cv2.VideoCapture(0)
cap.set(3, width)
cap.set(4, height)

while True:
    success, img = cap.read()
    img = cv2.flip(img, 1)

    cv2.imshow("image", img)
```

4.3. Memproses dan Mendapatkan Gestur Tangan

Citra gambar yang telah didapatkan menggunakan kamera akan diproses oleh program menggunakan *Mediapipe* yang telah dipermudah penggunaannya melalui *Library CVZone* untuk melakukan pengklasifikasian gestur tangan. Dari citra

yang telah diproses akan menghasilkan informasi landmark berupa bentuk tulang tangan. Kode yang digunakan adalah sebagai berikut:

```
detector = HandDetector(detectionCon=0.8, maxHands=2)

while True:
    success, img = cap.read()
    img = cv2.flip(img, 1)

    hands, img = detector.findHands(img, flipType=False)

    if hands:
        hand = hands[0]

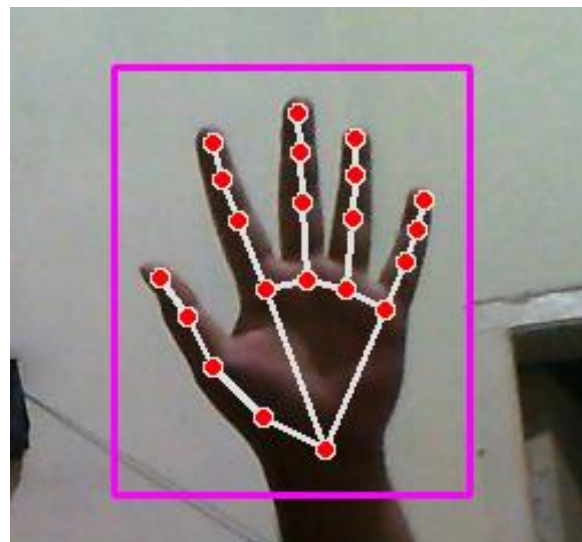
        fingers = detector.fingersUp(hand)

        print(fingers)

        cv2.imshow("image", img)
        key = cv2.waitKey(1)

    if key == ord('q'):
        break
```

Pada kode tersebut, terdapat penerapan *library Mediapipe* pada *package CVZone* yang digunakan untuk membantu mengklasifikasi gestur tangan, berupa informasi jari tangan yang sedang diangkat atau tidak diangkat. Hal ini memungkinkan program untuk membedakan antara satu gestur tangan dengan gestur tangan lainnya. Selain itu, fitur yang berfungsi untuk keluar dari program yang sedang berjalan juga telah ditambahkan. Dengan adanya kedua fungsi ini, program diharapkan mampu mendeteksi berbagai kondisi gestur tangan serta memberikan opsi untuk menghentikan eksekusi program sesuai yang dibutuhkan.



Gambar 4. Landmark Tangan Kiri Yang Terdeteksi

4.4. Menentukan Perintah Yang Akan Dieksekusi

Posisi *landmark* tangan akan diidentifikasi melalui penggunaan *mediapipe*, yang tersedia dalam perpustakaan *CVZone*. Posisi *landmark* tangan tersebut akan dijadikan referensi untuk mengeksekusi

perintah oleh komputer. Gestur tangan akan diartikan sebagai perintah tombol *keyboard* yang sesuai, seperti menggunakan panah kanan untuk beralih ke *slide* berikutnya, panah kiri untuk beralih ke *slide* sebelumnya, serta masuk/keluar dari mode layar penuh. Berbagai gestur tangan yang akan diterapkan telah dijabarkan pada Tabel 1.

Tabel 1. Landmark Perintah Gestur Tangan

No	Gestur Tangan	Perintah
1		Berpindah ke <i>slide</i> selanjutnya
2		Berpindah ke <i>slide</i> sebelumnya
3		Masuk ke mode layar penuh
4		Keluar dari mode layar penuh

Semua gestur tangan yang akan digunakan untuk diekstrak untuk mendapatkan informasi *landmark* tangan terdapat pada Tabel 1. Implementasi fungsi dari *package CVZone* akan diterapkan pada kode untuk memberikan kondisi sebagai acuan terhadap perintah yang akan di eksekusi. Adapun kode yang digunakan untuk melakukan implementasi tersebut, yaitu:

```
timeTracked = time.time()
gest = ""

if hands:
    hand = hands[0]

    fingers = detector.fingersUp(hand)

    if time.time() >= timeTracked + gestureCooldown and
    cy <= gestureThreshold:

        if fingers == [1, 1, 0, 0, 0]:
            pyautogui.press('right')
            print("Pressing Right.")
            gest = 'Next'

        elif fingers == [0, 0, 0, 0, 0]:
            pyautogui.press('left')
```

```
print("Pressing Left.")
gest = 'Previous'

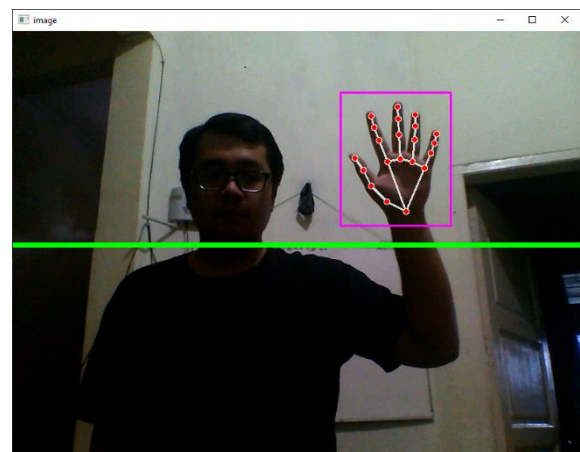
elif fingers == [1, 1, 1, 1, 1]:
    pyautogui.press('f5')
    print("Go to SlideShow.")
    gest = 'Fullscreen'

elif fingers == [1, 1, 1, 0, 0]:
    pyautogui.press('esc')
    print("Escape SlideShow.")
    gest = 'Exit Fullscreen'

else:
    gest = ""

timeTracked = time.time()
```

Di sini, penulis menemukan masalah ketika program secara tidak sengaja mendeteksi gestur-gestur yang tidak diinginkan selama presentasi. Solusi untuk masalah ini dengan membuat zona *Threshold*, yaitu pembagian zona deteksi yang bertujuan agar hanya tangan yang berada di dalam zone tersebut yang dapat menjalankan perintah-perintah *keyboard*. Selain itu, penulis menambahkan variabel "timeTracked" untuk mengatur perintah sehingga dapat dijeda untuk mencegah penumpukan perintah dalam waktu yang bersamaan ketika dijalankan.



Gambar 5. Area Eksekusi Perintah Gestur Tangan Dibatasi oleh Garis Hijau

4.5. Melakukan Penerapan dan Uji Coba Prototype Program

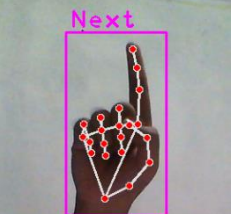
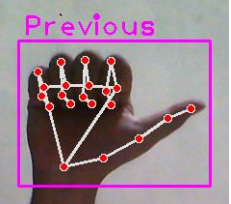
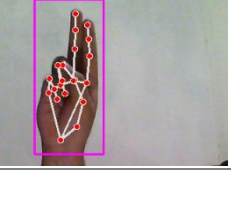
Pada IDE *PyCharm* yang digunakan, tidak terdapat *error* yang mengganggu eksekusi program *prototype* yang dirancang. Program juga telah diuji coba dengan menjalankan dan digunakan langsung terhadap dengan aplikasi *PowerPoint*. Disini program diuji coba dalam dua aspek, yaitu pengujian keberhasilan program dalam mendeteksi gestur tangan dan estimasi waktu bagi sistem untuk mendeteksi gestur tangan pada jarak tertentu. Pengujian ini akan dilakukan menggunakan perangkat yang memiliki spesifikasi sebagai berikut.

Tabel 2. Spesifikasi Perangkat Pengujian

Processor	Intel core i5-3317U CPU @ 1.70GHz (4CPUs), ~1.7GHz
RAM	12288 MB
Graphic Card	Intel HD 4000
VRAM (Approx. Total Memory)	1792 MB

Tahap pertama pada pengujian yang dilakukan berupa dengan menjalankan *prototype* dan menggunakannya secara langsung. Hal ini dilakukan untuk mendapatkan informasi keberhasilan program dalam mendeteksi gestur tangan untuk memberikan perintah. Gesur ini juga diharuskan untuk dapat menjalankan perintah yang akan diproses oleh program untuk mengoperasikan aplikasi *PowerPoint* dengan sesuai.

Tabel 3. Pengujian Keberhasilan Pendeteksian Gestur Tangan

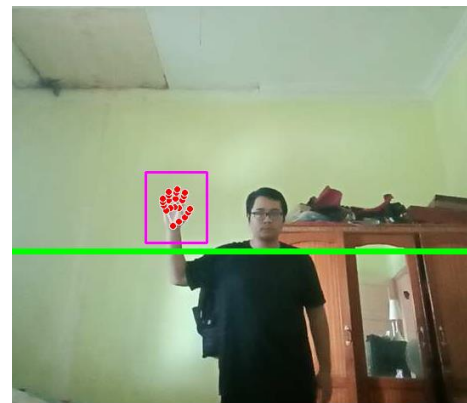
Perintah	Gestur Tangan	Hasil Pengujian
Berpindah ke <i>slide</i> selanjutnya		Berhasil
Berpindah ke <i>slide</i> sebelumnya		Berhasil
Masuk ke mode layar penuh		Berhasil
Keluar dari mode layar penuh		Berhasil

Dari hasil pengujian yang telah dijabarkan pada Tabel 3, diketahui bahwa program mampu untuk melakukan kontrol terhadap aplikasi *PowerPoint* dalam berbagai perintah yang sesuai. Perintah-perintah ini berupa melakukan pendeteksian gestur tangan untuk mengganti ke *slide* selanjutnya, mengganti ke *slide* sebelumnya, masuk mode layar penuh dan keluar mode layar penuh slide presentasi *PowerPoint*. Langkah selanjutnya adalah melakukan pengujian waktu responsif *prototype* yang telah

dirancang. Pengujian ini bertujuan untuk memperoleh informasi mengenai waktu responsif program berdasarkan jarak dan waktu tertentu. *Stopwatch* merupakan alat yang digunakan dalam pengukuran waktu responsif program ini.



Gambar 6. Citra Pengujian Pada Jarak 1 Meter



Gambar 7. Citra Pengujian Pada Jarak 2 Meter

Pengujian ini terdiri dari empat bagian yaitu menguji kemampuan deteksi tangan untuk berpindah ke *slide* selanjutnya, berpindah ke *slide* sebelumnya, masuk mode layar penuh, dan keluar dari mode layar penuh, serta mencantumkan jarak untuk setiap gestur tangan pada jarak 1 dan 2 meter. Hasil pengujian tersebut dijabarkan pada Tabel 4, Tabel 5, Tabel 6, dan Tabel 7.

Tabel 4. Hasil Percobaan Deteksi Tangan Untuk Berpindah Ke *Slide* Selanjutnya

BERPINDAH KE SLIDE SELANJUTNYA		
PERCOBAAN	1M	2M
1	4.18	1.46
2	1.57	2.35
3	1.27	2.23
4	1.94	2.58
5	4.64	2.41
6	1.23	2.01
7	2.29	1.1
8	2.53	1.18
9	1.46	1.26
10	1.74	2.07
Rata-rata	2.285	1.865
Total rata-rata	2.075 detik	

Tabel 5. Hasil Percobaan Deteksi Tangan Untuk berpindah Ke Slide Sebelumnya

BERPINDAH KE SLIDE SEBELUMNYA		
PERCOBAAN	1M	2M
1	1.63	1.24
2	1.19	1.12
3	1.62	2.63
4	1.74	2.29
5	1.98	1.53
6	2.3	3.01
7	1.27	1.13
8	1.14	2.33
9	1.94	1.37
10	1.78	1.29
Rata-rata	1.659	1.794
Total rata-rata	1.7265 detik	

Tabel 6. Hasil Percobaan Deteksi Tangan Untuk Masuk ke Mode Layar Penuh

MASUK MODE LAYAR PENUH		
PERCOBAAN	1M	2M
1	1.03	2.41
2	1.42	2.68
3	1.98	1.09
4	1.21	1.32
5	2.72	1.7
6	1.18	3.86
7	1.47	1.87
8	1.21	4.38
9	1.33	3.14
10	2.23	2.38
Rata-rata	1.578	2.483
Total rata-rata	2.0305 detik	

Tabel 7. Hasil Percobaan Deteksi Tangan Untuk Keluar Mode Layar Penuh

KELUAR MODE LAYAR PENUH		
PERCOBAAN	1M	2M
1	1.55	1.28
2	0.86	1.17
3	0.87	2.05
4	2.31	3.01
5	1.11	1.85
6	1.77	3.08
7	2.37	2.07
8	2.15	1.77
9	1.19	3.09
10	1.24	2.54
Rata-rata	1.542	2.191
Total rata-rata	1.8665 detik	

4.6. Penulisan Laporan

Dari pengujian *prototype* yang telah dilakukan, didapatkan informasi bahwa program telah berhasil dirancang dengan mengimplementasikan penggunaan bahasa pemrograman *Python* dengan *library OpenCV* untuk melakukan pemrosesan video, *Mediapipe* yang berperan sebagai *framework* untuk mendapatkan *landmark* tangan, dan *CVZone* untuk mempermudah melakukan klasifikasi gestur tangan. Program yang dirancang juga terbukti mampu untuk melaksanakan perintah dan berfungsi sesuai dengan yang diharapkan

dan mampu melakukan kontrol terhadap aplikasi *PowerPoint*.

Didapatkan hasil uji coba penggunaan *prototype* yang dilakukan bahwa untuk berpindah ke *slide* selanjutnya memerlukan waktu responsif rata-rata 2.075 detik, berpindah ke *slide* sebelumnya mendapatkan rata-rata 1.7265 detik, masuk mode layar penuh mendapatkan waktu rata-rata 2.0305 detik, dan keluar mode layar penuh mendapatkan waktu rata-rata 1.8665 detik. Pengujian pada jarak 1 meter mendapatkan rata-rata dalam waktu 1,766 detik. Sedangkan pada jarak 2 meter program mampu mengenali gestur tangan dalam waktu 2,08325 detik.

5. KESIMPULAN DAN SARAN

Melalui analisis permasalahan, perancangan serta pengujian dari bagian-bagian sebelumnya, penulis dapat menyimpulkan bahwa pengimplementasian *hand gesture recognition* pada program *prototype* kontrol aplikasi *PowerPoint* menggunakan bahasa pemrograman *Python* dengan *library OpenCV*, *MediaPipe* dan *CVZone* telah berhasil dirancang. *Prototype* ini telah diuji coba dan terbukti berhasil melakukan kontrol terhadap aplikasi *PowerPoint*. Program ini juga telah diuji coba pada jarak tertentu sehingga menghasilkan nilai estimasi responsif program dalam mengenali gestur tangan. Pada pengujian yang telah dilakukan didapatkan bahwa untuk berpindah ke *slide* selanjutnya memerlukan waktu responsif rata-rata 2.075 detik, berpindah ke *slide* sebelumnya mendapatkan rata-rata 1.7265 detik, masuk mode layar penuh mendapatkan waktu rata-rata 2.0305 detik, dan keluar mode layar penuh mendapatkan waktu rata-rata 1.8665 detik. Pengujian pada jarak 1 meter mendapatkan rata-rata dalam waktu 1,766 detik. Sedangkan pada jarak 2 meter program mampu mengenali gestur tangan dalam waktu 2,08325 detik. Dengan demikian, hasil pengujian dari program yang dirancang terbukti mampu untuk melaksanakan perintah dan berfungsi sesuai dengan yang diharapkan dan dapat memudahkan melakukan kontrol terhadap aplikasi *PowerPoint*.

DAFTAR PUSTAKA

- [1] G. I. Asmara, N. Erdina, R. A. Ariza, and Suhairi, "Urgensi Pertemuan dan Presentasi Dalam Organisasi Bisnis," *Dawatuna J. Commun. Islam. Broadcast.*, vol. 1, no. 2, pp. 109–119, 2021, doi: 10.47476/dawatuna.vii2.487.
- [2] M. Muthoharoh, "Media PowerPoint dalam Pembelajaran," vol. 26, no. 1, pp. 21–32, 2019.
- [3] M. Idrees, A. Ahmad, M. A. Butt, and H. M. Danish, "CONTROLLING POWER POINT USING HAND GESTURES IN PYTHON," *Webology*, vol. 18, no. 6, p. 2021, 2021, [Online]. Available: <http://www.webology.org><http://www.webology.org>
- [4] H. Yunita and E. Setyati, "HAND GESTURE

- RECOGNITION SEBAGAI PENGANTI MOUSE KOMPUTER MENGGUNAKAN KAMERA,” *J. ELTIKOM*, vol. 3, no. 2, pp. 64–76, 2019, doi: 10.31961/eltikom.v3i2.114.
- [5] I. Monisa, A. Z. Hasibuan, R. Aulia, and M. S. Asih, “Prototype Alat Pengontrol Pergerakan Tripod Berbasis Mikrokontroler,” *Algoritma. J. Ilmu Komput. dan Inform.*, vol. 5, no. 2, pp. 125–133, 2021.
- [6] 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.*, pp. 1–10, 2021, doi: 10.1088/1757-899x/1045/1/012043.
- [7] G. Sung *et al.*, “On-device Real-time Hand Gesture Recognition”, doi: 10.32628/cseit2173132.
- [8] L. Rahmawati, I. R. Efendi, and I. Umami, “Prototyping Virtual Mouse Menggunakan Gestur Tangan Berbasis Machine Learning,” *J. Sains Komput. Inform.*, vol. 6, no. 2, pp. 1002–1010, 2022.
- [9] Indriani, M. Harris, and A. S. Agoes, “Applying Hand Gesture Recognition for User Guide Application Using MediaPipe,” *Proc. 2nd Int. Semin. Sci. Appl. Technol. (ISSAT 2021) Appl.*, vol. 207, pp. 101–108, 2021, doi: 10.2991/aer.k.211106.017.
- [10] M. Peral, A. Sanfeliu, and A. Garrell, “Efficient Hand Gesture Recognition for Human-Robot Interaction,” 2022, doi: 10.1109/LRA.2022.3193251.
- [11] M. Mohamad, M. Y. M. Saman, and M. S. Hitam, “A Review on OpenCV,” pp. 1–17, 2015, doi: 10.13140/RG.2.1.2269.8721.
- [12] R. Test, L. Test, Y. Huang, I. Fietze, J. M. Cori, and J. E. Manousakis, “In Cabin Driver Monitoring and Alerting System For Passenger cars using Machine Learning,” *J. Phys. Conf. Ser.*, pp. 1–10, 2023, doi: 10.1088/1742-6596/2601/1/012040.
- [13] W. Fredi, P. Heri, I. K. Weny, and Musayyanah, “KONTROL KIPAS ANGIN SECARA JARAK JAUH MELALUI PENGENALAN BENTUK GESTUR JARI TANGAN BERBASIS COMPUTER VISION,” *J. Innov. Res. Knowl.*, vol. 3, no. 3, pp. 793–806, 2023.
- [14] I. P. Tambun, P. S. Matematika, and U. N. Malang, “KLASIFIKASI ALFABET BAHASA ISYARAT INDONESIA (BISINDO) DENGAN,” pp. 158–163, 2023.
- [15] F. Ramadhani, A. Satria, and Salamah, “Implementasi Algoritma Convolutional Neural Network dalam Mengidentifikasi Dini Penyakit pada Mata Katarak,” *sudo J. Tek. Inform.*, vol. 2, no. 4, pp. 167–175, 2023, doi: <https://doi.org/10.56211/sudo.v2i4.408>.
- [16] A. A. Himawan, S. Aulia, and M. Iqbal, “Rancang Bangun Ai Virtual Mouse Berbasis Pengolahan Citra,” *e-Proceeding Appl. Sci.*, vol. 9, no. 1, pp. 232–246, 2023, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/appliedscience/article/view/19409>
- [17] A. L. S. Saabith, T. Vinothraj, and M. M. M. Fareez, “A review on Python libraries and Ides for Data Science,” *Int. J. Res. Eng. Sci.*, vol. 09, no. 11, pp. 36–53, 2021, [Online]. Available: https://www.ijres.org/papers/Volume-9/Issue-11/Ser-2/G09113653.pdf%0Ahttps://www.researchgate.net/profile/Vinothraj-Thangarajah/publication/357898994_A_Review_on_Python_Libraries_and_IDEs_for_Data_Science/links/620249344d89183b338b49c2/A-Review-on-Python-
- [18] A. Triono, A. S. Budi, R. Abdillah, and Wahyudi, “Implementasi Peretasan Sandi Vigenere Chipper Menggunakan Bahasa Pemrograman Python,” *J. Multidisiplin Ris.*, vol. 1, no. 1, pp. 1–9, 2023, [Online]. Available: <https://jurnal.ittc.web.id/index.php/jumri>
- [19] I. Irwanto and G. G. Giantika, “Penyuluhan Penggunaan Kamera HP untuk Kebutuhan Kalangan Wanita RW 13 Babelan Bekasi,” *J. Abdi Masy. Indones.*, vol. 2, no. 3, pp. 833–842, 2022, doi: 10.54082/jamsi.334.
- [20] M. Oudah, A. Al-Naji, and J. Chahl, “Hand Gesture Recognition Based on Computer Vision: A Review of Techniques,” *J. Imaging*, vol. 6, no. 8, 2020, doi: 10.3390/JIMAGING6080073.
- [21] A. M. Chalik, B. A. Qowy, F. Hanafi, and Ahlijati Nuraminah, “Mouse Tracking Tangan dengan Klasifikasi Gestur Menggunakan OpenCV dan Mediapipe,” *J. Ilm. Tek. Inform. dan Komun.*, vol. 1, no. 2, pp. 10–18, 2021, [Online]. Available: <http://journal.sinov.id/index.php/juitik/article/view/323%0Ahttps://journal.sinov.id/index.php/juitik/article/download/323/250>