Controlling the Dji Ryze Tello Drone Using Human Hand Gestures

Widodo[1]*, Aldis S Andoy[2], Rosalin T Tayane[3], Iyus Supriadi[4]
Universitas Sains Teknologi Jayapura
Corresponding Author: Widodo widodommsi67@gmail.com

ARTICLEINFO

Keywords: Computer Vision, Hand Gestures Recognition, Drone DJI Ryze Tello

Received : 25 December
Revised : 25 January
Accepted: 25 February

ABSTRACT

Computer vision is an important part of the development of modern technology. One of the highlights is the introduction of hand movement, which integrates artificial intelligence into device controls. In the use of devices, especially drones or robots, this technology allows control without the need for additional devices such as remote control. Objectives Research applied technology Gesture Identification Hand drone DJI Ryze Tello. This application involves using Python 3 as a programming language, PyCharm as an editor, the OpenCV Library and Google MediaPipe (Hands) for real-time image processing and hand tracking. Research results show that drone control through hand gestures in front of a webcam can be done successfully, highlighting the potential of CVs to improve user interfaces on modern devices.

DOI Prefix: https://doi.org/10.59890/ijarss.v2i2.1417
ISSN-E: **3026-0205**
https://journal.multitechpublisher.com/index.php/ijarss/index

INTRODUCTION

Technology is evolving rapidly in the world of science and technology. One of the most important innovations in human interaction with computers is computer vision technology, which allows humans to interact with the computer without direct contact. In the field of computer vision, research focuses on hand gesture recognition technology. (HGR) (Azis et al., 2022)

Often, the control of drones or robots requires supporting devices such as remote control. However, some users are looking for natural input equipment without direct contact. In the modern era, to get advanced input equipment, users have to pay more. Case study using the affordable Dji Ryze Tello drone and supporting HGR technology through its SDK (Alamsyah, 2022)(Kurnia Rahman et al., 2019).

The research aims to detect hand signals by applying computer vision technology (HGR) to control the drone. (Studi Kasus: Dji Ryze Tello). Using an affordable and programmable drone, users can control it with just a hand gesture in front of a Webcam. This technology allows users to control the drone in a new and exciting way, replacing the remote controller's function.(Kurniawan et al., 2023). Research into Computer Vision (Hand Gestures Recognition) in controlling drones offers a number of significant advantages. First of all, this technology allows users to control drones just by using hand signals in front of the camera. This minimizes the need for additional devices such as remote control, providing a simpler and more natural user experience.

Another advantage lies in this technology's ability to improve accessibility. Drones used in research, such as Dji Ryze Tello, have SDKs that support the development of Hand Gestures Recognition (HGR). This makes it easier for users to implement this technology on drones at a more affordable cost compared to specialized devices or more expensive drones.

In addition, Computer Vision (Hand Gestures Recognition) deals with complex input equipment problems. Users can control the drone with just hand movements in front of webcam cameras, eliminating the need for often complex and expensive physical devices. This makes this technology very relevant in the context of more efficient drone flights..

When applied to drones, Hand Gestures Recognition provides high flexibility and mobility. Users can control the drone remotely without the need for additional devices that are difficult to carry. This is especially useful in situations where the use of conventional remote controls becomes impractical.

In the context of security, this technology also provides an advantage. Users can control the drone without touching physical devices, reducing the risk of damage or loss of control equipment. Moreover, this security aspect can be applied to a variety of drone uses, including surveillance, air recording, or other hobby activities.

Overall, this study reveals that the application of computer vision (hand gesture recognition) in controlling drones brings a number of advantages, including simplicity, affordability, and improved accessibility, as well as providing flexibility and significant security aspects.

LITERATURE REVIEW

Helda Yunita's research, "Hand Gesture Recognition as a Replacement for a Computer Mouse Using a Camera," uses cameras, convexhull algorithms, and hand color markers. The results showed that the convexhull method can detect hand objects well and recognize hand signals according to command with an accuracy of 68% out of 75%.(Yunita & Setyati, 2019).

A study written by Damatraseta, Novariany, and Ridhani in 2021 titled "Real-time Hand Gesture Detection and Identification with Deep Learning CNN" uses the HGD approach, which uses skin color division: CIELab color space and Convolutional Neural Networks. (CNN). The results were pretty good, with good model data: Table 2 showed that 26 letters of BISINDO were successfully predicted correctly, but it was necessary to perform several trials on each letter. Some letters, like B, D, J, K, and P, are difficult to predict. This is probably because they have a letter shape similar to other letters. Therefore, the HGD shape must be tested to obtain bias and weight values appropriate to its class. (Damatraseta et al., 2021).

Kusuma and Shodiq conducted research under the title "Intelligent Presentation Systems Using Hand Movement Identification Based on Signal Classification Electromyography (EMG)." Devices and application components used include myo watches, EMG data, feature extraction, classification methods, and intelligent presentation systems. As a result, an ANN model has been constructed to recognize EMG signals from hand movements. This model uses 32 input nodes, two hidden layers, a learning ratio of 0.1, a reverse propagation training method, a sigmoid activation function, and three outputs. The proposed model showed an accuracy of 81.67%, an average precision of 85.16%, and a recall of 83.66%. (Kusuma & Shodiq, 2018).

The study, written by Stelani Makahaube, entitled "Implementation of Gesture Identification Technology in the Self-Service Educational Framework," uses the OpenCV library, Mediapipe, webcam, and self-service gateway appliance. The results of research recognition gestures can be consistent with the purposes of research. Using a virtual mouse to read hand-held objects on a webcam without direct contact (Makahaube et al., 2021)

METHODOLOGY

a. Research Stream

Drone controls are conducted systematically and have consecutive stages, but the drone control stages conducted in this study can be seen in Figure 1.

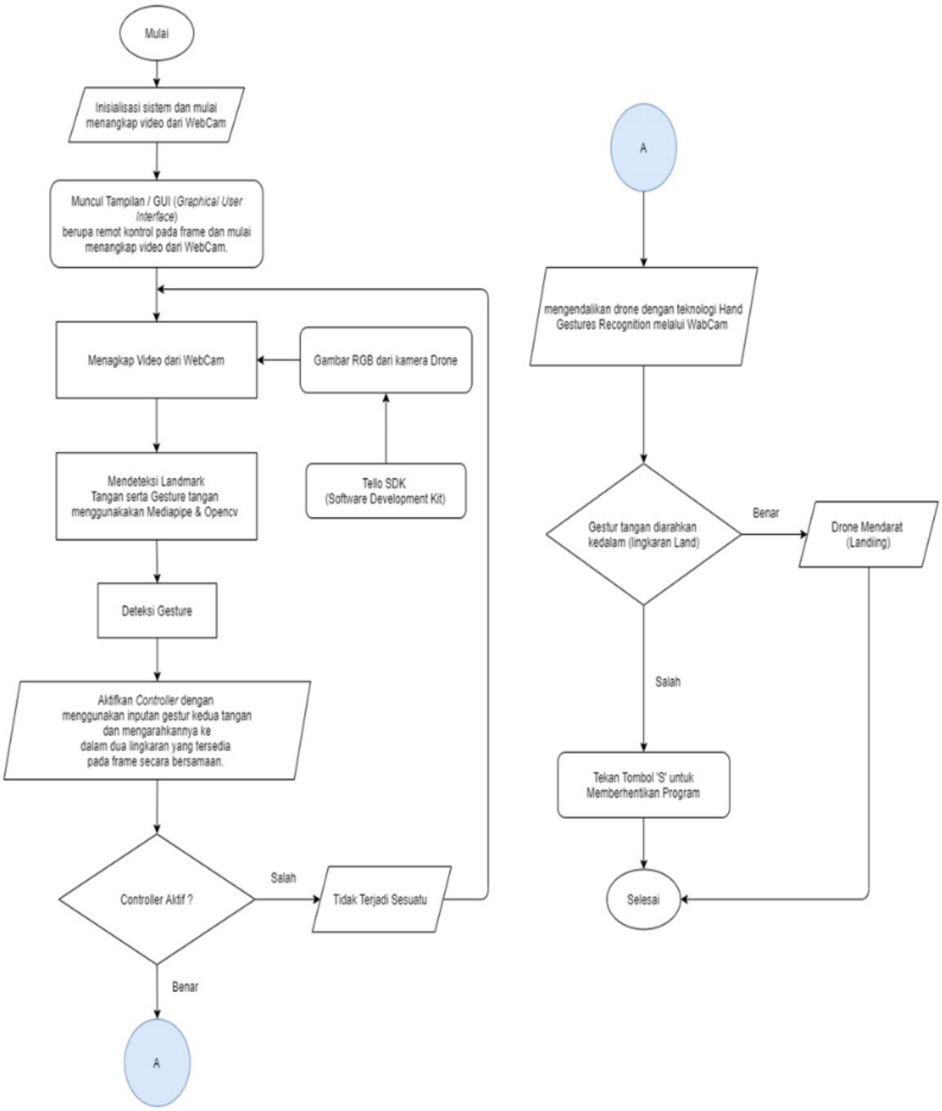

Mulai
Inisialisasi sistem dan mulai menangkap video dari WebCam
Muncul Tampilan / GUI (*Graphical User Interface*) berupa remot kontrol pada frame dan mulai menangkap video dari WebCam.
Menagkap Video dari WebCam
Gambar RGB dari kamera Drone
Tello SDK (Software Development Kit)
Mendeteksi Landmark Tangan serta Gesture tangan menggunakakan Mediapipe & Opencv
Deteksi Gesture
Aktifkan *Controller* dengan menggunakan inputan gestur kedua tangan dan mengarahkannya ke dalam dua lingkaran yang tersedia pada frame secara bersamaan.
Controller Aktif ?
Salah
Tidak Terjadi Sesuatu
Benar
A
A
mengendalikan drone dengan teknologi Hand Gestures Recognition melalui WabCam
Gestur tangan diarahkan kedalam (lingkaran Land)
Benar
Drone Mendarat (Landiing)
Salah
Tekan Tombol 'S' untuk Memberhentikan Program
Selesai

Figure 1. Drone control circuit

b. System Architecture

The system is designed to control drones with Hand Gesture Identification (HGR) technology, which allows users to control the drone through hand gesture recognition. As for an overview of the architecture of the system, you can see it in Figure 2.

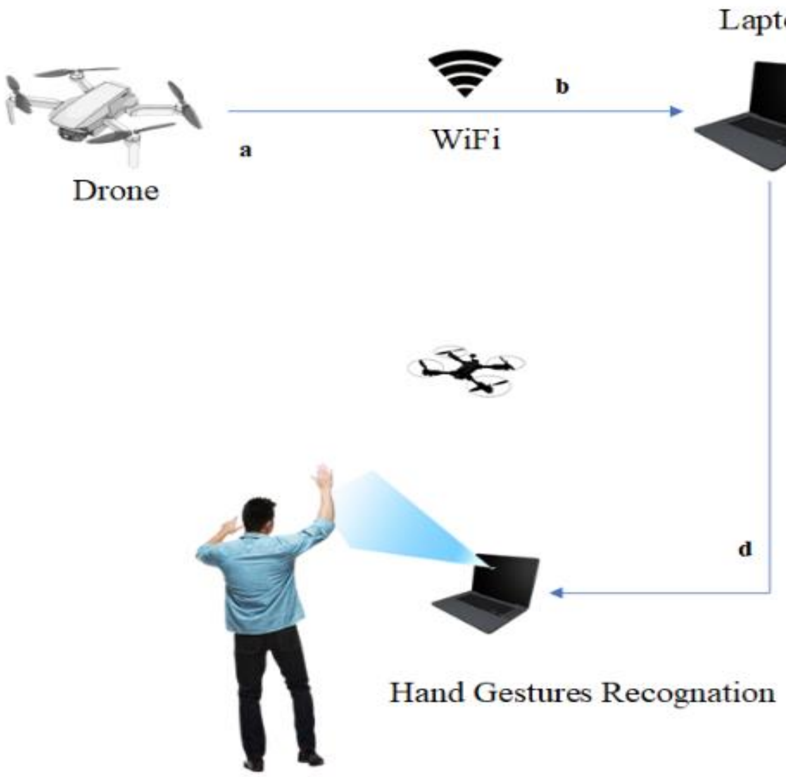


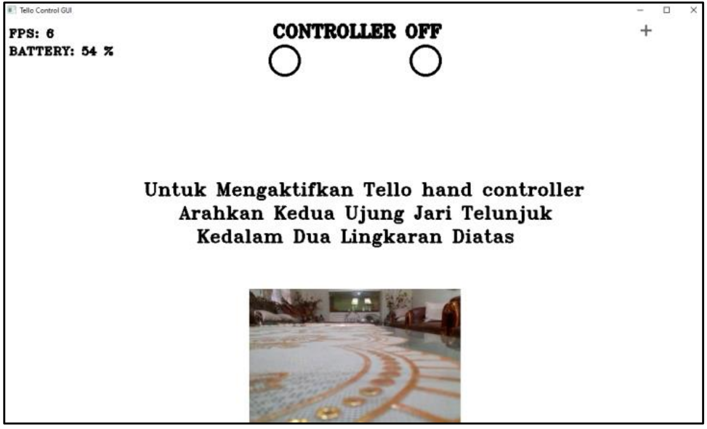
Figure 2. System Architecture

Based on figure 2, the user activates the drone to be used. Next, the user connects the drone to the laptop via a WiFi connection. Then the user runs the program on the laptop and performs hand calibration. And finally, the user controls the drone using Hand Gesture Recognition (HGR) technology via the webcam.

RESEARCH RESULT

A. Application Controller Off

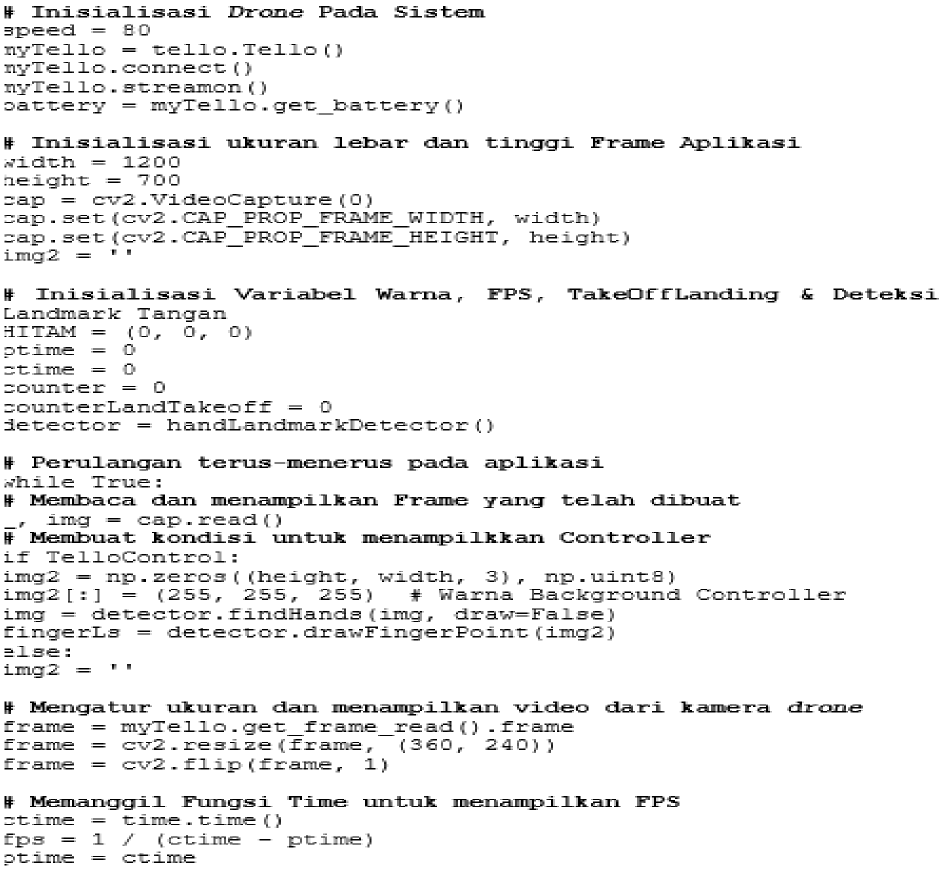
This view represents the uatama display that will appear when the drone is running and the controller is still off status that can be seen in Figure 3.



Untuk Mengaktifkan Tello hand controller
Arahkan Kedua Ujung Jari Telunjuk
Kedalam Dua Lingkaran Diatas

Figure 3. Control Off

To control the drone in off mode use the following command:



```
# Inisialisasi Drone Pada Sistem
speed = 80
myTello = tello.Tello()
myTello.connect()
myTello.streamon()
battery = myTello.get_battery()

# Inisialisasi ukuran lebar dan tinggi Frame Aplikasi
width = 1200
height = 700
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, width)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, height)
img2 = ''

# Inisialisasi Variabel Warna, FPS, TakeOffLanding & Deteksi
Landmark Tangan
HITAM = (0, 0, 0)
ptime = 0
ctime = 0
counter = 0
counterLandTakeoff = 0
detector = handLandmarkDetector()

# Perulangan terus-menerus pada aplikasi
while True:
# Membaca dan menampilkan Frame yang telah dibuat
_, img = cap.read()
# Membuat kondisi untuk menampilkkan Controller
if TelloControl:
img2 = np.zeros((height, width, 3), np.uint8)
img2[:] = (255, 255, 255)  # Warna Background Controller
img = detector.findHands(img, draw=False)
fingerLs = detector.drawFingerPoint(img2)
else:
img2 = ''

# Mengatur ukuran dan menampilkan video dari kamera drone
frame = myTello.get_frame_read().frame
frame = cv2.resize(frame, (360, 240))
frame = cv2.flip(frame, 1)

# Memanggil Fungsi Time untuk menampilkan FPS
ctime = time.time()
fps = 1 / (ctime - ptime)
ptime = ctime
```

B. Control On Application

Control On will appear when the drone is running and the controller is activated by pointing both thumbs into the two available circles, then automatically the ontroller is on or active. Users can use the same method to disable the controllers. Which can be seen in Figure 4.

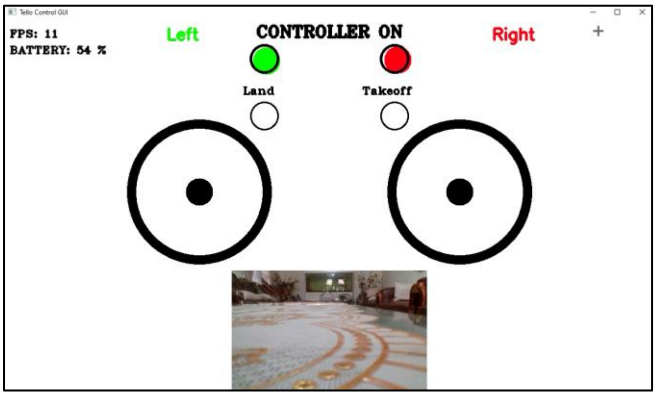
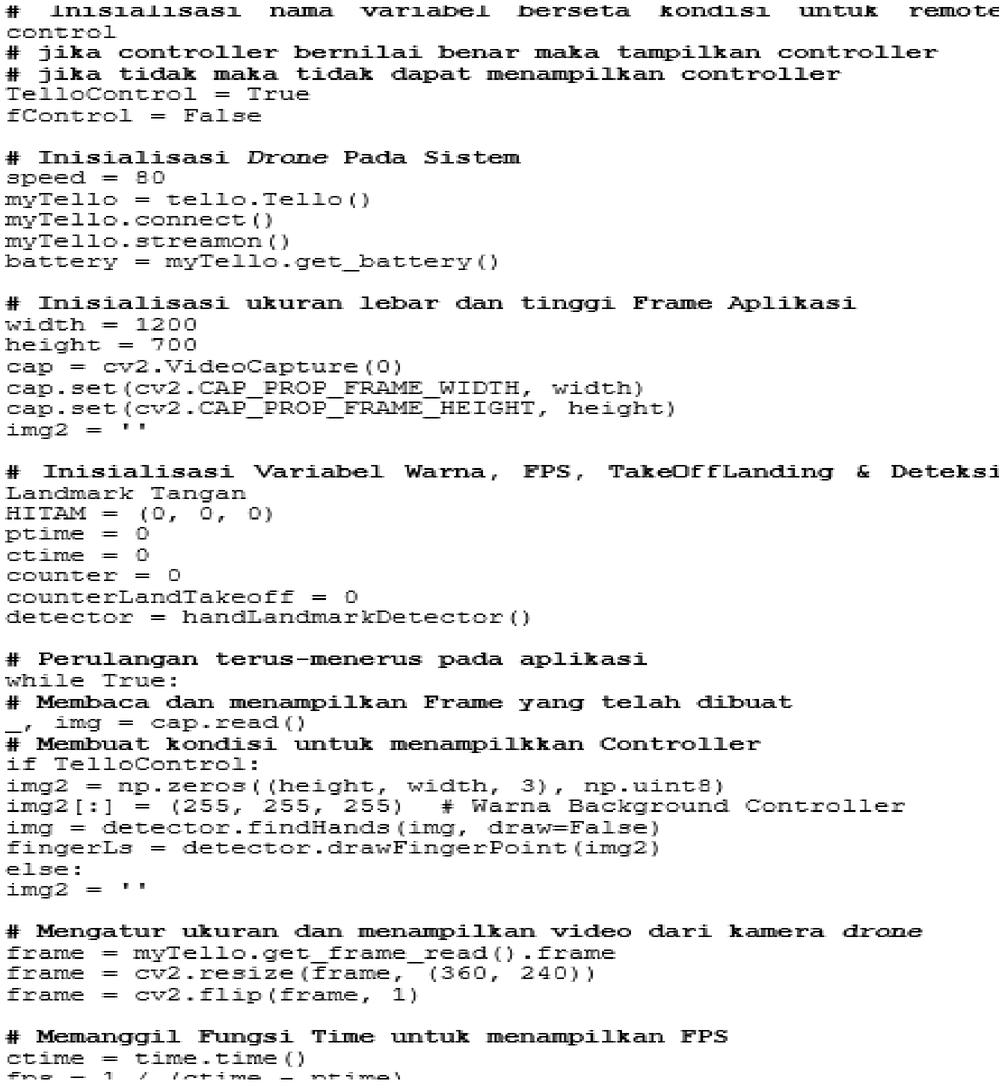


Figure 4. Controller On

Drone commands on mode on are as follows:



```
# Inisialisasi nama variabel berseta kondisi untuk remote
control
# jika controller bernilai benar maka tampilkan controller
# jika tidak maka tidak dapat menampilkan controller
TelloControl = True
fControl = False

# Inisialisasi Drone Pada Sistem
speed = 80
myTello = tello.Tello()
myTello.connect()
myTello.streamon()
battery = myTello.get_battery()

# Inisialisasi ukuran lebar dan tinggi Frame Aplikasi
width = 1200
height = 700
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, width)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, height)
img2 = ''

# Inisialisasi Variabel Warna, FPS, TakeOffLanding & Deteksi
Landmark Tangan
HITAM = (0, 0, 0)
ptime = 0
ctime = 0
counter = 0
counterLandTakeoff = 0
detector = handLandmarkDetector()

# Perulangan terus-menerus pada aplikasi
while True:
# Membaca dan menampilkan Frame yang telah dibuat
_, img = cap.read()
# Membuat kondisi untuk menampilkkan Controller
if TelloControl:
img2 = np.zeros((height, width, 3), np.uint8)
img2[:] = (255, 255, 255)  # Warna Background Controller
img = detector.findHands(img, draw=False)
fingerLs = detector.drawFingerPoint(img2)
else:
img2 = ''

# Mengatur ukuran dan menampilkan video dari kamera drone
frame = myTello.get_frame_read().frame
frame = cv2.resize(frame, (360, 240))
frame = cv2.flip(frame, 1)

# Memanggil Fungsi Time untuk menampilkan FPS
ctime = time.time()
fps = 1 / (ctime - ptime)
```

C. Control Take Off

The controller has been on and the user will takeoff by pointing the finger from the right hand into the takeoff circle to fly the drone that can be seen in Figure 5.

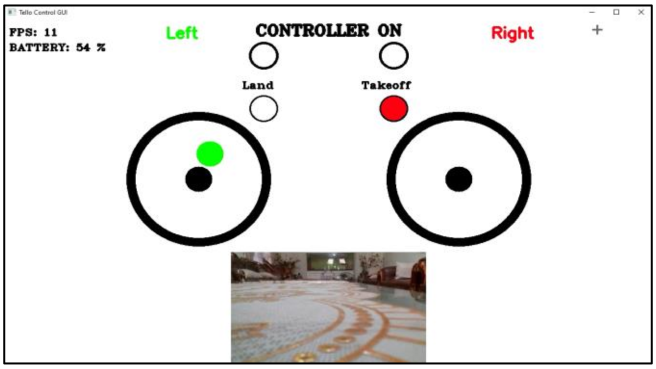


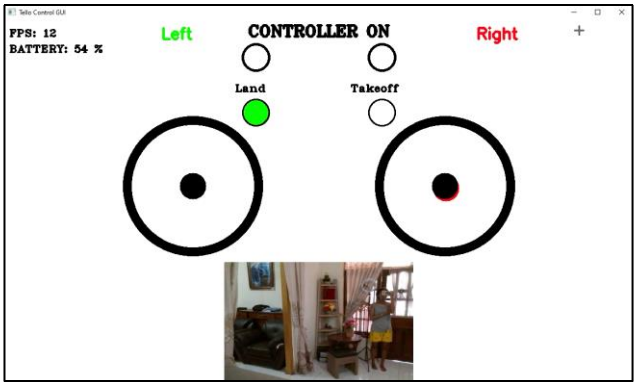
Figure 5. Takeoff Controller

The command to control takeoff on this drone is as follows.

```
# Inisialisasi nama variabel berseta kondisi untuk remote
control
# jika controller bernilai benar maka tampilkan controller
# jika tidak maka tidak dapat menampilkan controller
TelloControl = True
fControl = False

# Inisialisasi Drone Pada Sistem
speed = 80
myTello = tello.Tello()
myTello.connect()
myTello.streamon()
battery = myTello.get_battery()

# Inisialisasi ukuran lebar dan tinggi Frame Aplikasi
width = 1200
height = 700
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, width)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, height)
img2 = ''

# Inisialisasi Variabel Warna, FPS, TakeOffLanding & Deteksi
Landmark Tangan
HITAM = (0, 0, 0)
ptime = 0
ctime = 0
counter = 0
counterLandTakeoff = 0
detector = handLandmarkDetector()

# Perulangan terus-menerus pada aplikasi
while True:
# Membaca dan menampilkan Frame yang telah dibuat
_, img = cap.read()
# Membuat kondisi untuk menampilkkan Controller
if TelloControl:
img2 = np.zeros((height, width, 3), np.uint8)
img2[:] = (255, 255, 255)  # Warna Background Controller
img = detector.findHands(img, draw=False)
fingerLs = detector.drawFingerPoint(img2)
else:
img2 = ''

# Mengatur ukuran dan menampilkan video dari kamera drone
frame = myTello.get_frame_read().frame
frame = cv2.resize(frame, (360, 240))
frame = cv2.flip(frame, 1)
```

D. Control Landing

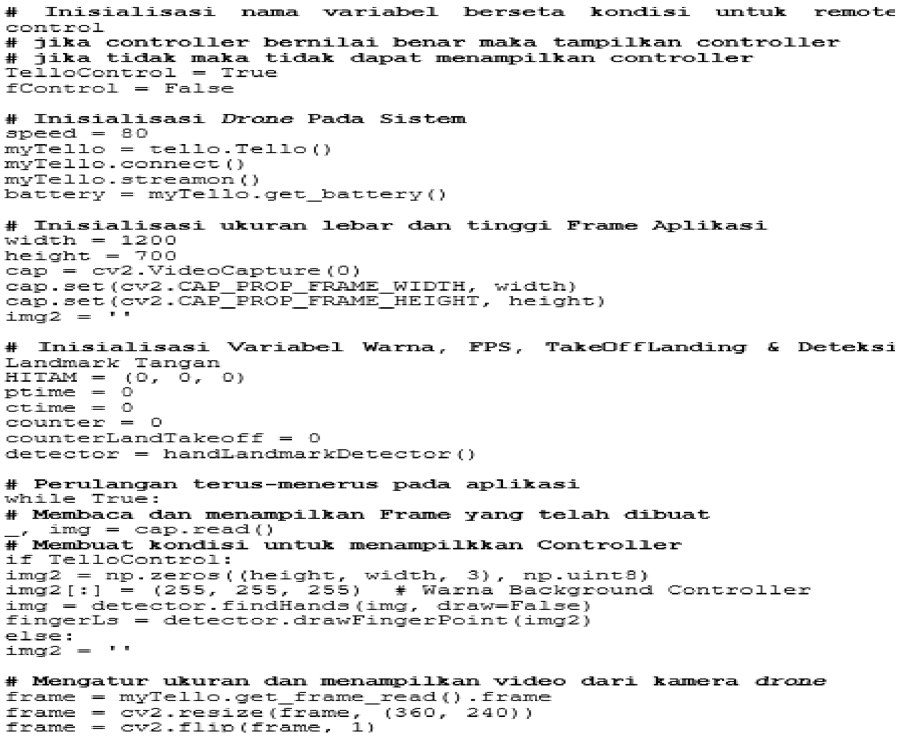
The controller is on, and the user will land by pointing the index finger from the right hand into the landing circle to land the drone, which can be seen in Figure 6.



FPS: 12
BATTERY: 54 %

Figure 6. Controler Landing

As for the drone command on the landing menu is as follows:



```
# Inisialisasi nama variabel berseta kondisi untuk remote
control
# jika controller bernilai benar maka tampilkan controller
# jika tidak maka tidak dapat menampilkan controller
TelloControl = True
fControl = False

# Inisialisasi Drone Pada Sistem
speed = 80
myTello = tello.Tello()
myTello.connect()
myTello.streamon()
battery = myTello.get_battery()

# Inisialisasi ukuran lebar dan tinggi Frame Aplikasi
width = 1200
height = 700
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, width)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, height)
img2 = ''

# Inisialisasi Variabel Warna, FPS, TakeOffLanding & Deteksi
Landmark Tangan
HITAM = (0, 0, 0)
ptime = 0
ctime = 0
counter = 0
counterLandTakeoff = 0
detector = handLandmarkDetector()

# Perulangan terus-menerus pada aplikasi
while True:
# Membaca dan menampilkan Frame yang telah dibuat
_, img = cap.read()
# Membuat kondisi untuk menampilkkan Controller
if TelloControl:
img2 = np.zeros((height, width, 3), np.uint8)
img2[:] = (255, 255, 255)  # Warna Background Controller
img = detector.findHands(img, draw=False)
fingerLs = detector.drawFingerPoint(img2)
else:
img2 = ''

# Mengatur ukuran dan menampilkan video dari kamera drone
frame = myTello.get_frame_read().frame
frame = cv2.resize(frame, (360, 240))
frame = cv2.flip(frame, 1)
```

E. Testing

In this study have been performed several forms of testing faced drones to find out the accuracy and functionality of instrument control and have obtained the results described in table 1.

Table 1. Testing Result

Distance	Values	Strength
DISTANCE	0 – 25 M	At a distance of 0-25 meters, the drones can be controlled well. The remainder of that distance then there will be instability in controlling the drone, due to the limited connectivity distance of the WiFi.
HIGHLIGHTS	0 – 10 M	At an altitude of 0-6 meters, the drone can still be controlled but if it exceeds that distance then there will be instability in controlling the drones.
HAND AND WEBCAM	0 – 3.5	At a distance of 0-3.5 meters, hands can still be detected well, beyond that range then the Webcam can't detect hand gestures.

CONCLUSIONS AND RECOMMENDATIONS

1. Using computer vision technology to control a drone, users can experience the most advanced features of the drone in a more natural way by using motion or hand gestures directly in front of the webcam without having to use remote controls.
2. The system created during the test has some shortcomings, such as: a. The drone can only be controlled if the user is right before the webcam because the application has a remote-control display. b. Webcams can

detect and recognize gestures of hands up to a maximum distance of 3.5 meters.

3. Users cannot control the drones directly through the camera device. Features or modes for controlling drones with keyboards are not available. e. An application that indicates the distance of the drone from the ground surface is unavailable. Poor lighting in the environment can interfere with the performance of the drone and make it difficult to identify hand gestures.

ADVANCED RESEARCH

For the development of a system or application, it is necessary to add features that can perform face and hand tracking, which will help users interact with the drone; features or modes to help users control the drones directly through the camera of the device; features and modes for controlling drones using a keyboard; and features that provide information about the distance between the drone and the ground surface.

REFERENCES

Alamsyah, N. (2022). Rancang Bangun Dan Implementasi Sistem Kendali Robot Penanggulangan Bencana Alam Pantai Angin Mammiri Makassar. *Bulletin of Information Technology (BIT)*, *3*(4), 407–412. https://journal.fkpt.org/index.php/BIT/article/view/434

Azis, N., Azzahra, A. S., Muditomo, A., Medika, U. A., & Timur, J. (2022). Analysis Of Human Computer Interaction Approach In Pospay Application. *Jurnal Mantik*, *6*(36), 1956–1963.

Damatraseta, F., Novariany, R., & Ridhani, M. A. (2021). Real-time BISINDO Hand Gesture Detection and Recognition with Deep Learning CNN. *Jurnal Informatika Kesatuan*, *1*(1), 71–80. https://doi.org/10.37641/jikes.v1i1.774

Kurnia Rahman, A., Supriyanto, H., & Meizinta, T. (2019). RANCANG BANGUN DAN IMPLEMENTASI SISTEM KENDALI QUADCOPTER MELALUI JARINGAN INTERNET BERBASIS LOKASI MENGGUNAKAN SMARTPHONE ANDROID. *Seminar Nasional Kontrol, Instrumentasi Dan Otomasi (SNIKO) 2018*, 307–318. https://doi.org/10.5614/sniko.2018.35

Kurniawan, M. A. S., Arap, N. A., Irawan, A., & Azizah, N. (2023). *Digitalisasi Pendidikan Berbasis Teknologi Abad 21 (AI , AR , VR , Iot , Blockchain , Drones , Gamification , Machine Learning , Robotics , 3D Printing)*. *3*(3), 230–241.

Kusuma, D. H., & Shodiq, M. N. (2018). Sistem Presentasi Cerdas Menggunakan Pengenalan Gerakan Tangan Berdasarkan Klasifikasi Dari Sinyal Electromyography (EMG) Menggunakan Myo Armband. *INTENSIF*, *2*(1), 36. https://doi.org/10.29407/intensif.v2i1.11939

Makahaube, S. S., Sambul, A. M., & Sompie, S. R. (2021). Implementation of Gesture Recognition Technology for Automated Education Service Kiosk. *Jurnal Teknik Informatika*, *16*(4), 1–8. https://doi.org/https://doi.org/10.35793/jti.16.4.2021.34210

Yunita, H., & Setyati, E. (2019). Hand Gesture Recognition Sebagai Pengganti Mouse Komputer Menggunakan Kamera. *Jurnal ELTIKOM*, *3*(2), 64–76. https://doi.org/10.31961/eltikom.v3i2.114