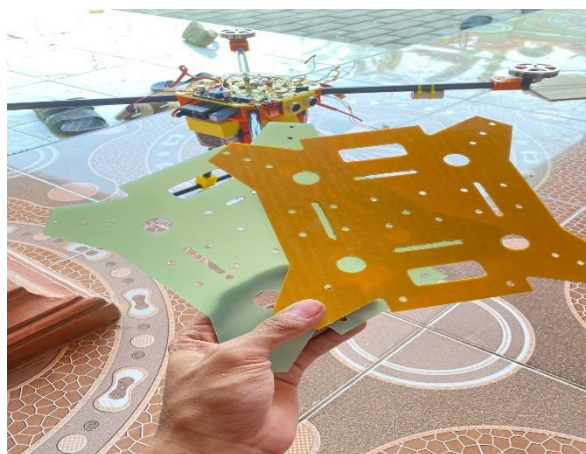


LAMPIRAN

DAFTAR LAMPIRAN

Lampiran 1. Dokumentasi Pembuatan Drone



Lampiran 2. Form Pembimbing 1

FORM BIMBINGAN








TUGAS AKHIR

NAMA : M. FARIS FALAHUDIN

NIM : 21010006

JUDUL TA : APLIKASI IMAGE PROCESING DRONE VTOL (VERTICAL TAKE-OFF AND LANDING) UNTUK PENGAMBILAN DAN PELETAKAN OBJEK BERBASIS RASPBERRY PI 4

Pembimbing 1

No	Hari/Tanggal	Uraian	Tanda tangan
1.	21 Maret 2024	Bimbingan Judul	
2.	28 Maret 2024	Bimbingan bab 1	
3.	1 April 2024	Revisi bab 1 dan bimbingan bab 2	
4.	23 April 2024	Revisi bab 2 dan bimbingan bab 3	
5.	2 Agustus	bimbingan bab 4 dan 5	
6.	10 Agustus	ACC + lampiran	 

Lampiran 3. Form Pembimbing 2

FORM BIMBINGAN









TUGAS AKHIR

NAMA : M. FARIS FALAHUDIN

NIM : 21010006

JUDUL TA : APLIKASI IMAGE PROCESING DRONE VTOL (VERTICAL TAKE-OFF AND LANDING) UNTUK PENGAMBILAN DAN PELETAKAN OBJEK BERBASIS RASPBERRY PI 4

Pembimbing 2

No	Hari/Tanggal	Uraian	Tanda tangan
1.	21 Maret 2024	Bimbingan judul	
2.	25 Maret 2024	Bimbingan bab 1	
3.	1 April 2024	Revisi bab 1 dan bimbingan bab 2	
4.	23 April 2024	Revisi bab 2 dan bimbingan bab 3	
5.	05 Juni 2024	Bimbingan bab 3	
6.	19 Juni 2024	Bimbingan flowchart dan wiring diagram	
7.	2 Agustus 2024	Bimbingan bab 4 dan bab 5	
8.	20 Agustus 2024	ACC	

Lampiran 4. Kesiediaan Pembimbing 1

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini:

Nama : Rony Darpono, M.T
NIPY : 09.015.282
Jabatan : Ka. Prodi DIII Teknik Elektronika

Dengan ini menyatakan bersedia untuk menjadi Pembimbing 1 pada Tugas Akhir mahasiswa berikut :

Nama : M. Faris Falahudin
NIPY : 21010006
Program Studi : DIII Teknik Elektronika
Judul Laporan Tugas Akhir : **APLIKASI IMAGE PROCESING
DRONE VTOL (VIRTUAL TAKE OFF
LANDING) UNTUK PENGAMBILAN
DAN PELETAKAN OBJEK BERBASIS
RASPBBERY PI 4**

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Mengetahui
Ka. Prodi DIII Teknik Elektronika



Rony Darpono M.T
NIPY.09.015.282

Tegal, 19 Maret 2024
Calon Dosen Pembimbing 1

Rony Darpono M.T
NIPY.09.015.282

Lampiran 5. Kesiadaan Pembimbing 2

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini:

Nama : Bahrun Niam, M.T
NIPY : 09.015.277
Jabatan : Sek. Prodi DIII Teknik Elektronika

Dengan ini menyatakan bersedia untuk menjadi Pembimbing 2 pada Tugas Akhir mahasiswa berikut :

Nama : M. Faris Falahudin
NIPY : 21010006
Program Studi : DIII Teknik Elektronika
Judul Laporan Tugas Akhir : **APLIKASI IMAGE PROCESING
DRONE VTOL (VIRTUAL TAKE OFF
LANDING) UNTUK PENGAMBILAN
DAN PELETAKAN OBJEK BERBASIS
RASPBBERY PI 4**

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Mengetahui
Ka. Prodi DIII Teknik Elektronika


Rony Darpono M.T
NIPY.09.015.282

Tegal, 19 Maret 2024
Calon Dosen Pembimbing 2



Bahrun Niam M.T.
NIPY.09.015.277

Lampiran 6. Penilaian Bimbingan

PENILAIAN BIMBINGAN TUGAS AKHIR INDIVIDU

Judul Tugas Akhir : Aplikasi *Image Procecing Drone VTOL (Vertical Take Off And Landing)* Untuk Pengambilan Dan Peltakan Objek Berbasis Raspberry Pi 4

Nama : M. Faris Falahudin
Nim : 21010006
Kelas : 6A

I. Nilai Bimbingan Tugas Akhir (Pembimbing I)

NO	Unsur Yang Dinilai	Nilai
1.	Kedisiplinan Dalam Bimbingan	95
2.	Kreatifitas Pemecahan Dalam Bimbingan	93
3.	Penguasaan Materi Tugas Akhir	94
4.	Kelengkapan Dan Referensi Tugas Akhir	92
Total Nilai = $\left(\frac{\text{jumlah nilai}}{4}\right)$		93,5


II. Nilai Bimbingan Tugas Akhir (Pembimbing II)

NO	Unsur yang dinilai	Nilai
1.	Kedisiplinan Dalam Bimbingan	90
2.	Kreatifitas Pemecahan Dalam Bimbingan	95
3.	Penguasaan Materi Tugas Akhir	95
4.	Kelengkapan Dan Referensi Tugas Akhir	90
Total Nilai = $\left(\frac{\text{jumlah nilai}}{4}\right)$		92,5

$$\text{Nilai Bimbingan} = \frac{\text{Total nilai pembimbing 1} + \text{Total nilai pembimbing 2}}{2} = \frac{93}{2} = \dots$$

Tegal, 21 Agustus 2024
Mengetahui,

Pembimbing I,

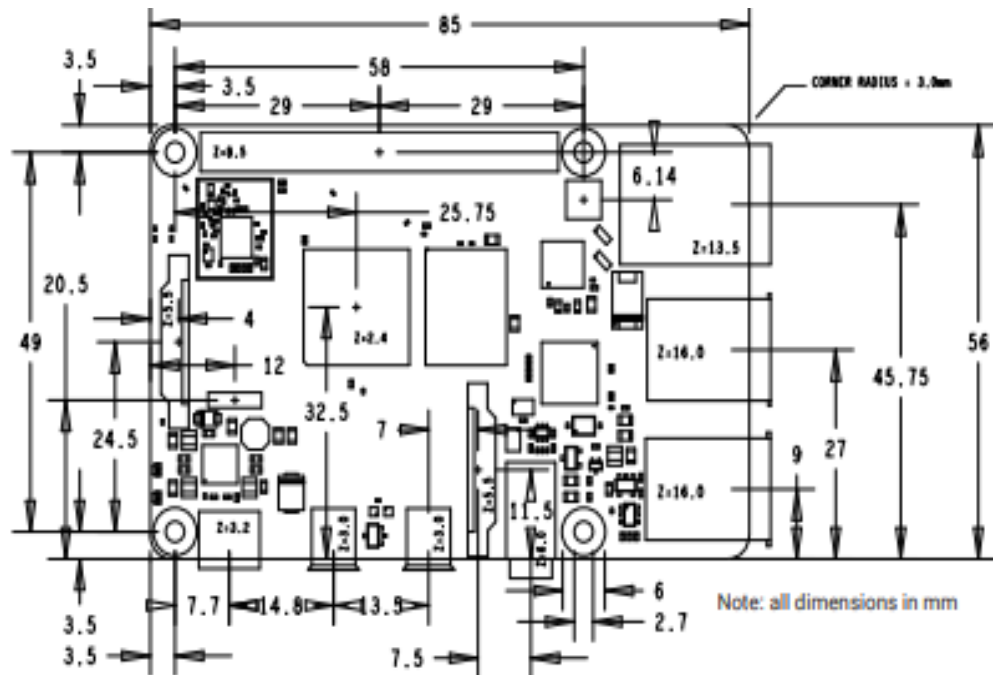

Rony Darpono M.T.

Pembimbing II,


Bahrun Niam M.T.

Lampiran 7. Datasheet Raspberry Pi 4 B+

Processor:	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory:	1GB, 2GB or 4GB LPDDR4 (depending on model)
Connectivity:	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports.
GPIO:	Standard 40-pin GPIO header (fully backwards-compatible with previous boards)
Video & sound:	2 × micro HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port
Multimedia:	H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 graphics
SD card support:	Micro SD card slot for loading operating system and data storage
Input power:	5V DC via USB-C connector (minimum 3A ¹) 5V DC via GPIO header (minimum 3A ¹) Power over Ethernet (PoE)-enabled (requires separate PoE HAT)
Environment:	Operating temperature 0–50°C
Compliance:	For a full list of local and regional product approvals, please visit https://www.raspberrypi.org/documentation/hardware/raspberrypi/conformity.md
Production lifetime:	The Raspberry Pi 4 Model B will remain in production until at least January 2026.



WARNINGS

- This product should only be connected to an external power supply rated at 5V/3A DC or 5.1V/ 3A DC minimum¹. Any external power supply used with the Raspberry Pi 4 Model B shall comply with relevant regulations and standards applicable in the country of intended use.
- This product should be operated in a well-ventilated environment and, if used inside a case, the case should not be covered.
- This product should be placed on a stable, flat, non-conductive surface in use and should not be contacted by conductive items.
- The connection of incompatible devices to the GPIO connection may affect compliance and result in damage to the unit and invalidate the warranty.
- All peripherals used with this product should comply with relevant standards for the country of use and be marked accordingly to ensure that safety and performance requirements are met. These articles include but are not limited to keyboards, monitors and mice when used in conjunction with the Raspberry Pi.
- Where peripherals are connected that do not include the cable or connector, the cable or connector must offer adequate insulation and operation in order that the relevant performance and safety requirements are met.

SAFETY INSTRUCTIONS

To avoid malfunction or damage to this product please observe the following:

- Do not expose to water, moisture or place on a conductive surface whilst in operation.
- Do not expose it to heat from any source; Raspberry Pi 4 Model B is designed for reliable operation at normal ambient room temperatures.
- Do not expose the printed circuit board to high-intensity light sources (e.g. xenon flash or laser) whilst in operation.
- Take care whilst handling to avoid mechanical or electrical damage to the printed circuit board and connectors.
- Avoid handling the printed circuit board whilst it is powered and only handle by the edges to minimise the

Lampiran 8. Datasheet Pixhawk 2.4.8

Processor:	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory:	1GB, 2GB or 4GB LPDDR4 (depending on model)
Connectivity:	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports.
GPIO:	Standard 40-pin GPIO header (fully backwards-compatible with previous boards)
Video & sound:	2 × micro HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port
Multimedia:	H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 graphics
SD card support:	Micro SD card slot for loading operating system and data storage
Input power:	5V DC via USB-C connector (minimum 3A ¹) 5V DC via GPIO header (minimum 3A ¹) Power over Ethernet (PoE)-enabled (requires separate PoE HAT)
Environment:	Operating temperature 0–50°C
Compliance:	For a full list of local and regional product approvals, please visit https://www.raspberrypi.org/documentation/hardware/raspberrypi/conformity.md
Production lifetime:	The Raspberry Pi 4 Model B will remain in production until at least January 2026.

Lampiran 9. Datasheet BM5010 Brushless

5010 360KV High Torque Brushless Motor for Drone

Features:

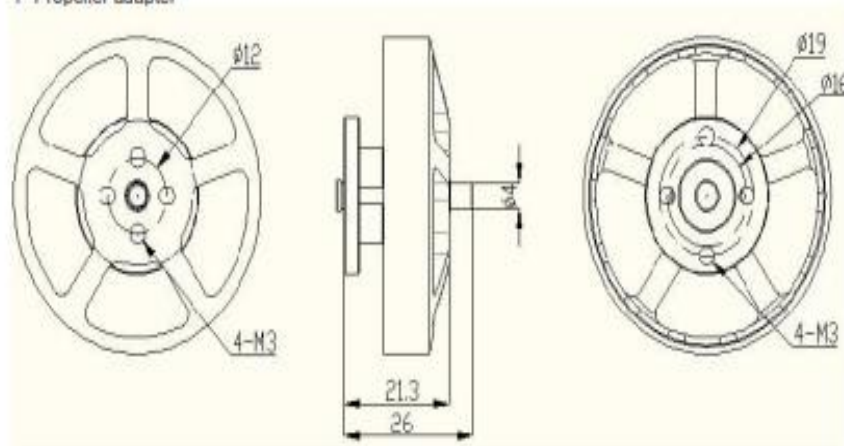
1. Machined aluminum front housing with four fan-style cooling holes that pump air through the motor while it runs.
2. Rear threaded mounting holes with both 16 mm and 19mm hole spacings fit a variety of applications.
3. Specially designed NdFeB magnets with high temperature rating for trouble-free operation.
4. High Temperature 140 C (356 F) rated wire is used for winding the motors to minimize the risk of burning up the motor.
5. High quality stator plates are epoxy coated on the inner surface to prevent winding shorts. 0.20mm stator plate id used on this motor 5010.
6. High quality shielded and permanently lubricated ball bearings are used to support the motor shaft in all our motors.
7. High temperature adhesives are used to secure the stator windings and prevent them from shifting and getting pinched or shorting out .

Spec:

KV: 360
Configu-ration: 12N14P
Shaft Diameter(out size): 4mm
Shaft Diameter(inside size): 5mm
Motor Dimension(Dia.*Len): $\Phi 50 \times 26$
Weight(g): 92g
Cable 40cm

Package:

1* 5010 motor
4* Screw
1* Propeller adapter



Lampiran 10. Coding

```
import time
import cv2
import numpy as np
from dronekit import connect, VehicleMode
from pymavlink import mavutil
import RPi.GPIO as GPIO
from pynput import keyboard

# Koneksi ke kendaraan
vehicle = connect('127.0.0.1:14550', wait_ready=True)
# Konfigurasi pin sensor ultrasonik
GPIO.setmode(GPIO.BCM)
TRIG_BOTTOM = 23
ECHO_BOTTOM = 24
TRIG_FRONT = 20
ECHO_FRONT = 21

GPIO.setup(TRIG_BOTTOM, GPIO.OUT)
GPIO.setup(ECHO_BOTTOM, GPIO.IN)
GPIO.setup(TRIG_FRONT, GPIO.OUT)
GPIO.setup(ECHO_FRONT, GPIO.IN)

# Fungsi untuk membaca jarak dari sensor ultrasonik
def read_ultrasonic(trig, echo):
    GPIO.output(trig, True)
    time.sleep(0.00001)
    GPIO.output(trig, False)

    start_time = time.time()
    stop_time = time.time()

    while GPIO.input(echo) == 0:
        start_time = time.time()

    while GPIO.input(echo) == 1:
        stop_time = time.time()

    time_elapsed = stop_time - start_time
    distance = (time_elapsed * 34300) / 2
    return distance

# Fungsi untuk mendeteksi warna tertentu
def detect_color(frame, lower_color, upper_color):
```

```

hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
mask = cv2.inRange(hsv, lower_color, upper_color)
mask_blurred = cv2.GaussianBlur(mask, (9, 9), 2)

contours, _ = cv2.findContours(mask_blurred, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
largest_contour = None
max_area = 0
for contour in contours:
    area = cv2.contourArea(contour)
    if area > max_area:
        max_area = area
        largest_contour = contour

if largest_contour is not None:
    M = cv2.moments(largest_contour)
    if M["m00"] != 0:
        cX = int(M["m10"] / M["m00"])
        cY = int(M["m01"] / M["m00"])
        cv2.drawContours(frame, [largest_contour], -1, (0, 255, 0), 2)
        cv2.circle(frame, (cX, cY), 7, (255, 255, 255), -1)
        return (cX, cY), frame
return None, frame

# Fungsi untuk mengatur pergerakan drone menggunakan channel overrides
def send_override(vehicle, pwm_roll, pwm_pitch, pwm_throttle, duration):
    vehicle.channels.overrides = {
        '1': pwm_roll,    # Roll (Kiri-Kanan)
        '2': pwm_pitch,   # Pitch (Maju-Mundur)
        '3': pwm_throttle # Throttle (Naik-Turun)
    }
    time.sleep(duration)
    vehicle.channels.overrides = {}

# Mengatur warna oranye untuk deteksi objek
lower_orange = np.array([5, 150, 150])
upper_orange = np.array([15, 255, 255])

# Mengatur resolusi kamera
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

# Variabel untuk menyimpan status tombol
key_pressed = None
exit_program = False # Menambahkan flag untuk keluar dari loop

```



```

# Fungsi untuk menangani penekanan tombol
def on_press(key):
    global key_pressed, exit_program
    try:
        key_pressed = key.char
        if key_pressed == 'x': # Jika tombol 'x' ditekan
            exit_program = True
    except AttributeError:
        pass

def on_release(key):
    global key_pressed
    key_pressed = None

# Memulai listener keyboard
listener = keyboard.Listener(on_press=on_press, on_release=on_release)
listener.start()

# Fungsi untuk takeoff ke ketinggian tertentu
def takeoff(target_altitude):
    print(f"Takeoff to {target_altitude} meter")
    vehicle.mode = VehicleMode("GUIDED")
    vehicle.armed = True
    vehicle.simple_takeoff(target_altitude)

    while True:
        current_altitude = vehicle.location.global_relative_frame.alt
        if current_altitude >= target_altitude * 0.95:
            print("Reached target altitude")
            break
        time.sleep(1)

# Fungsi utama untuk misi drone
def main_mission():
    takeoff(1)

    print("Moving forward 1 meter")
    send_override(vehicle, 1500, 1600, 1500, 5)

    while True:
        ret, frame = cap.read()
        if not ret:
            break

    if exit_program: # Mengecek flag keluar

```

```

        break

# Deteksi warna
color_center, frame = detect_color(frame, lower_orange, upper_orange)
cv2.imshow("Camera Feed", frame)

if color_center is not None:
    x, y = color_center
    frame_width = frame.shape[1]
    frame_height = frame.shape[0]

    # Jika objek berada di tengah-tengah layar
    if y > frame_height * 0.45 and y < frame_height * 0.55 and x >
frame_width * 0.45 and x < frame_width * 0.55:
        print("Object detected and centered, landing")

    # Mendarat
    vehicle.mode = VehicleMode("LAND")
    while vehicle.armed:
        time.sleep(0.1) # Tunggu hingga drone benar-benar mendarat
    print("Landed")

    # Tunggu setengah detik
    time.sleep(0.5)

    # Takeoff lagi ke ketinggian 1 meter
    takeoff(1)

    # Lanjutkan misi
    print("Moving forward 1 meter after takeoff")
    send_override(vehicle, 1500, 1600, 1500, 5)

    if key_pressed == 'x': # Mengecek tombol 'x' melalui variabel key_pressed
        break
    print("Landing")
    vehicle.mode = VehicleMode("LAND")

cap.release()
cv2.destroyAllWindows()

# Memulai misi utama
main_mission()

# Menutup koneksi ke kendaraan
vehicle.close()
GPIO.cleanup()

```

Lampiran 11. Turnitin

APLIKASI IMAGE PROCESING DRONE VTOL (VERTICAL TAKE-OFF AND LANDING) UNTUK PENGAMBILAN DAN PELETAKAN OBJEK BERBASIS RASPBERRY PI 4

ORIGINALITY REPORT



PRIMARY SOURCES

1	eprints.polsri.ac.id Internet Source	2%
2	digilib.unila.ac.id Internet Source	2%
3	eprints.unisla.ac.id Internet Source	2%
4	eprints.poltektegal.ac.id Internet Source	2%
5	repository.unj.ac.id Internet Source	2%
6	eprints.itenas.ac.id Internet Source	1%
7	ijcs.stmikindonesia.ac.id Internet Source	1%
8	radith-kiul.blogspot.com Internet Source	1%

www.coursehero.com