



DESAIN DAN IMPLEMENTASI QUADCOPTER PADA PENGAPLIKASIAN DROPPING KIT DENGAN MENGGUNAKAN METODE BLOB DETECTION

Quadcopter Design and Implementation in Dropping Kit Applications Using Blob Detection Method

¹⁾ Muhammad Dwi Paksi Pratama, ²⁾ Hari Kurnia Safitri, ³⁾ Yulianto

^{1,2,3} Politeknik Negeri Malang, Indonesia

*Email: ¹⁾ dwipaksi567@gmail.com, ²⁾ hari.kurnia@polinema.ac.id, ³⁾ yulianto_poltek@yahoo.com

*Correspondence: Muhammad Dwi Paksi Pratama

DOI:

10.59141/comserva.v3i07.1069

ABSTRAK

Quadcopter adalah sebuah wahana tanpa awak (*Unmanned Aerial Vehicle*) yang saat ini sedang dikembangkan dan sangat berguna dalam berbagai bidang. *Quadcopter* masih sulit untuk dioperasikan yaitu salah satunya menjaga agar tetap stabil saat menjalankan misi dalam Kontes Robot Terbang yaitu menjatuhkan kit. Sebagai langkah selanjutnya, metode yang akan digunakan *Blob detection* dengan menggunakan *library open CV* untuk mempermudah melakukan misi *dropping* dengan mendeteksi warna yang dibantu oleh *optical flow sensor* yang bertujuan untuk menstabilkan gerakan *roll* dan *pitch* dan juga sensor lidar yang bertujuan untuk mengatur jarak ketinggian *Quadcopter*. Semua pemrosesan tersebut diproses melalui GCS (*Ground Controll Station*) atau laptop secara otomatis dengan memanfaatkan kamera FPV sebagai objek pengambilan gambar. Hasil penelitian menunjukkan bahwa akurasi sensor lidar sebagai sensor ketinggian dari *Quadcopter* mampu mendeteksi blob diposisi ketinggian 2,5 meter dan hasil *error* dari lidar mendapatkan hasil *error* sebesar 15,94% dari total ketinggian mulai dari 0.5 meter sampai dengan 3 meter. Sehingga total akurasi dari keakuratan *Quadcopter* untuk melakukan *dropping* sebesar 84,06% dari ketinggian tersebut.

Kata Kunci: *Blob Detection, Dropping Kit, Kamera FPV (First Person View), Quadcopter*

ABSTRACT

Quadcopter is a type of unmanned aircraft which is currently being developed and is very useful in various fields. The quadcopter is still difficult to operate, one of which is to keep it stable while carrying out missions in the Flying Robot Contest, namely dropping kits. As a next step, the method that will be used is blob detection using the openCV library to make it easier to carry out dropping missions by detecting colors assisted by optical flow sensors that aim to stabilize roll and pitch movements and also lidar sensors that aim to adjust the Quadcopter's altitude distance. And all of this processing is automatically processed through the GCS (*Ground Controll Station*) or laptop by utilizing the FPV camera as the object for shooting. The results showed that the accuracy of the lidar sensor as a height sensor from the Quadcopter is able to detect blob at a height of 2.5 meters and the error results from the lidar get an error of 15.94% of the total height starting from 0.5 meters to 3 meters. So that the total accuracy of the accuracy of the Quadcopter for dropping is 84.06% from that height.

Keywords: *Blob Detection, Dropping Kit, Kamera FPV (First Person View), Quadcopter*

PENDAHULUAN

Masa saat ini UAV, lebih sering digunakan untuk keperluan *mapping*, *video recording*, pencarian objek memanfaatkan teknologi *artificial intelligence*, dan berbagai keperluan digital lainnya. Karena UAV memiliki kegunaan yang begitu luas dalam banyak aspek (Hochstenbach et al. 2015). UAV atau pesawat tanpa awak adalah suatu perangkat udara yang tidak memiliki awak di dalamnya dan dapat dikendalikan dari jarak jauh atau secara otomatis (Manan, T, and T 2020) menggunakan sistem komputer yang sudah deprogram sebelumnya atau yang biasa kita sebut dengan *drone* (Yang, You Yang, and Isi Huang Fu 2020). Wahana terbang UAV (*Unnamed Aerial Vehicle*) yang saat ini dipelajari adalah *Quadcopter* (Palaha, Teles, and Yolnasdi 2020).

Pertandingan kontes robot terbang yang dilakukan sebelumnya, belum mendapatkan hasil yang maksimal yaitu dalam sebuah misi dalam menstabilkan posisi ketika *Quadcopter* berada diatas target sebelum melakukan misi dropping sehingga pada penelitian ini berfokus pada desain *Quadcopter*, pergerakan, dan mengoptimalkan misi dropping (Hadi et al. 2014) dengan metode *blob detection*. Tujuan penelitian ini adalah memposisikan *Quadcopter* tepat di atas target yaitu berupa warna dan memposisikan kestabilan *Quadcopter* saat *take-off* menjalankan misi.

Vertical Take-off and Landing secara otomatis adalah salah satu misinya yang biasa diterapkan pada *Quadcopter* (Palaha et al. 2020) (Palaha et al. 2020). Dalam konteks ini, metode *Blob Detection* digunakan untuk mengidentifikasi dan memisahkan gumpalan objek dalam gambar. Pengenalan bentuk seperti lingkaran, kotak, dan lainnya dapat dipermudah dengan pemanfaatan library open CV sebagai alat bantu. (Pamungkas, Sumbodo, and Candradewi 2017)

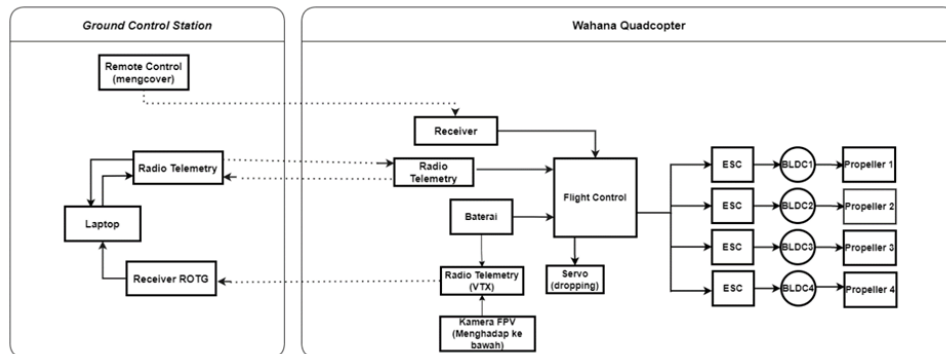
Pada penelitian ini dilakukan sistem *take-off* menuju ke target untuk melakukan dropping yang dirancang dan diimplementasikan pada *Quadcopter* dengan menggunakan metode deteksi blob. Metode tersebut dilakukan dengan mendeteksi warna yang diambil oleh kamera FPV yang nantinya akan digunakan sebagai visualisasi *waypoint* (Ramadhan, Yulianto, and Prasetyo Isnomo 2022) untuk men dropping kit dan digunakan sensor optik yang bertujuan untuk mempertahankan sudut *pitch and roll* yang membuat *Quadcopter* mampu mempertahankan posisi stabil diudara (*hover*) lalu melakukan *dropping*. Analisa pada penelitian ini yaitu dengan membaca ketinggian dari sensor LiDAR (Alamsyah and Rivai 2020) (Aziz and Zakarijah 2022) serta kestabilan *Quadcopter* untuk mengetahui keakuratan saat *dropping*.

METODE

Blok Diagram Sistem

Pixhawk yang berfungsi sebagai Flight control bertugas untuk menyeimbangkan wahana dengan bantuan lidar dan px4flow. Sensor optik yang terhubung ke *Flight Controller* melalui I2C yang mana sensor tersebut digunakan untuk mengunci gambar dibawah drone agar drone bisa melayang stabil, dan terdapat sensor lidar yang mengukur ketinggian *Quadcopter* terhubung melalui serial 4 pada *Flight Controller*. Terdapat 4 motor BLDC sebagai penggerak drone yang di kendalikan oleh Flight Controller dengan bantuan ESC. Baterai men -*supply Flight Control* dan kamera. Flight Controller pada sistem ini berkomunikasi dengan ground control station (Laptop) secara nirkabel menggunakan radio *telemetry*, untuk metode pengambilan gambar di lakukan oleh kamera FPV yang di kirim langsung ke *ground control* (Laptop) secara nirkabel yang kemudian diolah menggunakan *library vision* untuk mendapatkan koordinat, koordinat tersebut diolah menggunakan *python* yang nilai *output* di sesuaikan

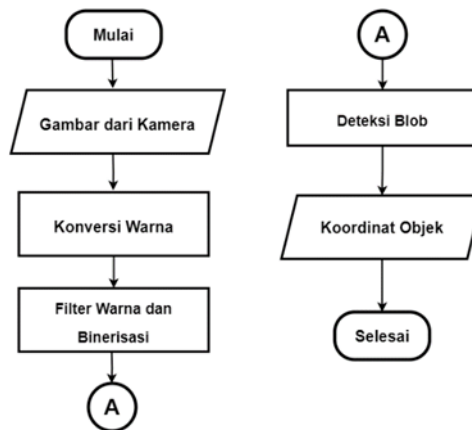
untuk penggerak *Quadcopter*, dan di kirim ke *Flight Controller* secara nirkabel dari ground control station (Laptop). Komunikasi data antara camera dan pixhawk dilakukan melalui ROS Connection. Wahana dikontrol dan dimonitor melalui PC/Laptop melalui koneksi wireless. Pada sistem ini Quadcopter tidak hanya dioperasikan secara terprogram, tetapi umumnya dapat dikendalikan oleh *remote control* seperti pada umumnya, dikarenakan untuk alasan keamanan jika drone gagal menyelesaikan misinya atau kehilangan kendali.



Gambar 1. Blok Diagram Sistem

Blob Detection

Dalam lingkup penelitian ini, penggunaan metode deteksi blob menjadi pilihan utama yaitu dengan mencari kumpulan pixel yang sama dan terbesar, yang sebelumnya sudah di filter dan dibinerisasikan, dalam pengolahan citra tersebut menggunakan program *library* OpenCV, dikarenakan penyederhanaan program terkait citra digital oleh sebab itu dapat mempercepat pengerjaan.

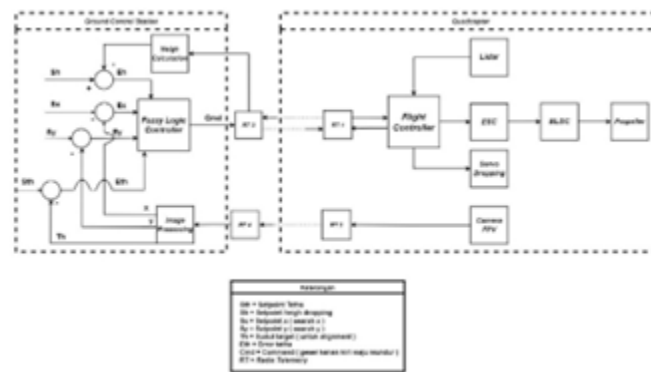


Gambar 2. Flowchart Diagram Blob

Blok Diagram Kontrol

Ketika *Quadcopter* distart (menggunakan *remote controller*), *Quadcopter* akan diperintah naik oleh *fuzzy* pada ketinggian tertentu dan atau mendekati ke arah target. Jika target telah terlihat (terdeteksi)

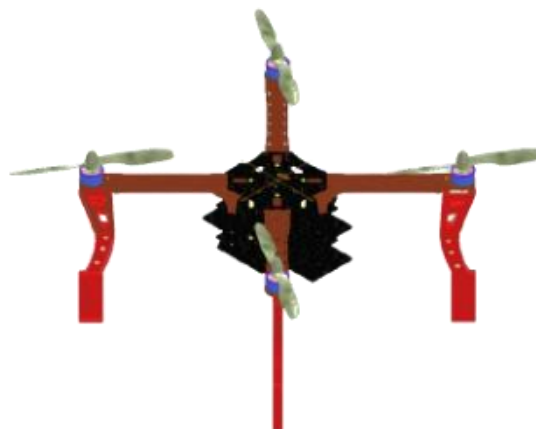
pada kamera yaitu nilai x dan y hasil IP ada nilainya atau tidak bernilai kosong, maka kontroler *fuzzy* akan melakukan proses fuzzifikasi, inferensi dan defuzzifikasi untuk menghasilkan sinyal *command* (cmd). Sebagai contoh jika jarak x atau y masih terlalu jauh maka *command* yang dihasilkan *fuzzy* berupa geser kanan dan kiri (sebanding dengan error e_x dan e_y) akan dikirim ke *Quadcopter* melalui RT (Radio Telemetri). Proses ini akan diulang terus menerus hingga didapatkan e_x dan e_y yang lebih kecil dari pada toleransi tertentu. Jika kondisi ini telah tercapai dan fitur *alignment* diaktifkan maka *fuzzy* bagian *alignment* akan berfungsi dengan output *fuzzy* berupa *command* putar kanan atau kiri. Jika proses *alignment* telah selesai, selanjutnya *fuzzy* akan berpindah ke mode kontrol ketinggian dengan *setpoint* sh. Dalam mode ini sistem kontrol akan mengukur jarak *Quadcopter* ke ground menggunakan lidar. Jika jarak terlalu tinggi maka *fuzzy* akan mengirim *command* turun ke *Quadcopter* hingga jarak terukur sama atau mendekati nilai sh. Jika kondisi ketinggian telah memenuhi syarat maka kontroler *fuzzy* mengirim *command* pelepasan target.



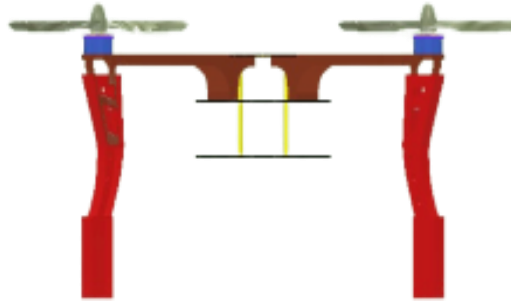
Gambar 3. Blok Diagram Control

Perancangan Mekanik

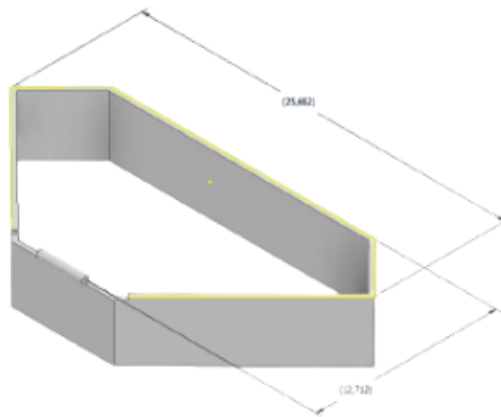
Berikut merupakan keseluruhan model desain *Quadcopter* pada penelitian ini seperti pada gambar berikut:



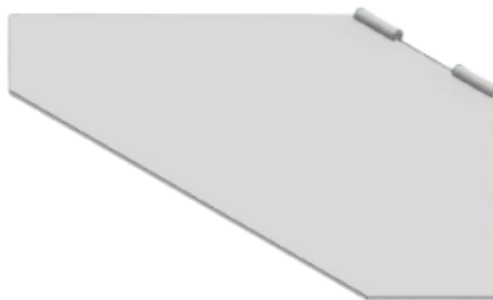
Gambar 4. Kerangka Quadcopter



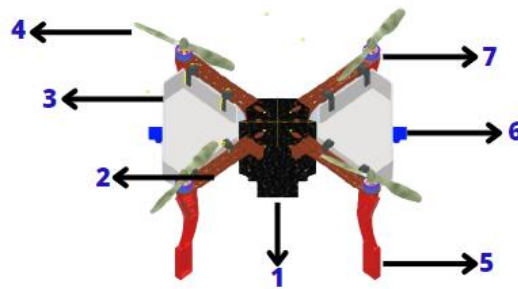
Gambar 5. Tampak Depan



Gambar 6. Desain Box



Gambar 7. Desain Pintu



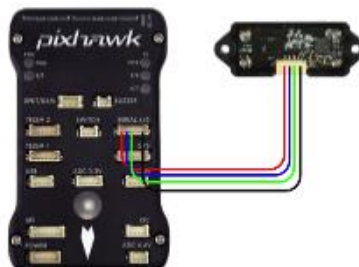
Gambar 8. Quadcopter Assembly Box Dropping

Perancangan Elektrik

Berikut menunjukkan koneksi dari rangkaian elektronik yang digunakan pada Quadcopter. Diantara yaitu *Flight Controller* dengan *Ground Control Station* (laptop) yang terhubung menggunakan Radio Telemetry, komunikasi ini menggunakan frekuensi yang sama yaitu 433 mHz (Palaha et al. 2020). Data yang didapat dari Pixhawk ke laptop yaitu meliputi data status penerbangan, orientasi. Sensor LiDAR yang berfungsi untuk mengukur jarak *Quadcopter* dari permukaan. Selanjutnya PX4 yang membantu menstabilkan *Quadcopter* pada saat *take-off* dan kemudian **kamera FPV berfungsi meng-capture gambar dan akan diproses di Ground Control Station (Laptop).**



Gambar 9. Koneksi Pixhawk ke Laptop



Gambar 10. Koneksi LiDAR ke Pixhawk



Gambar 11. Koneksi PX4 ke Pixhawk



Gambar 12. Koneksi Kamera ke GCS

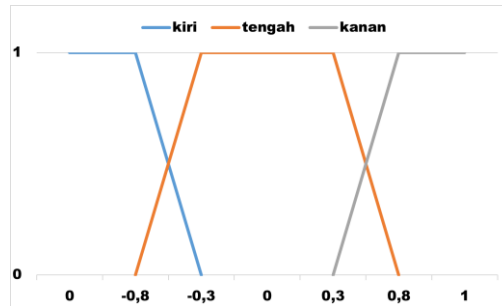
Perancangan Software

Pada bagian ini mempunyai beberapa fungsi yang yaitu memerintahkan drone bergerak *vertical* ke atas dengan kecepatan dan ketinggian yang sudah di atur dengan bantuan sensor, selanjutnya drone hover atau melayang dengan mengatur ke-tiga channel dengan nilai yang sama.

```
print("Hover")
import time
timestr = time.strftime("%Y%m%d-%H%M%S")
timestr2 = datetime.now()
with open("Hover_"+timestr+".csv", "w") as f:
    writer = csv.writer(f)
    writer.writerow(['Pith', 'Roll'])
    i = 0
    while i<80:
        vehicle.channels.overrides[3] = 1500
        vehicle.channels.overrides[1] = 1500
        vehicle.channels.overrides[2] = 1500
        time.sleep(0.20)
        if i%1==0:
            print("y : "+str(vehicle.attitude.pitch)+" || x : "+str(vehicle.attitude.roll)+" Time: "+str(i/5))
            writer.writerow([str(vehicle.attitude.pitch), str(vehicle.attitude.roll)])
            i = i+1
```

Perancangan Sistem Kontrol

1) Input *fuzzy* Horizontal



Gambar 13. Himpunan *Fuzzy* Horizontal

Tabel 1: Input *Fuzzy* Horizontal

Fungsi	Nama Variablel	Keanggotaan
X Axis	Kiri	<-0.8 -0.8 -0.3
	Tengah	-0.8 0 0.8
	Kanan	0.3 0.8 >0.8

Untuk memperoleh nilai membership digunakan rumus berikut :

Rentang nilai : $(-0.8) \leq \text{Pointer} \leq (-0.3)$ cm (Kategori Kiri/Kurva Turun)

$$\begin{aligned} \mu_b &= (c - \text{Pointer}) / (c - b) \\ &= ((-0.3) - \text{Pointer}) / ((-0.3) - (-0.8)) \\ &= ((-0.3) - \text{Pointer}) / (-0.5) \\ &= 0.6 - 1.5\text{Pointer} \end{aligned}$$

Rentang nilai : $(-0.8) \leq \text{Pointer} \leq (-0.3)$ cm (Kategori Tengah/Kurva Naik)

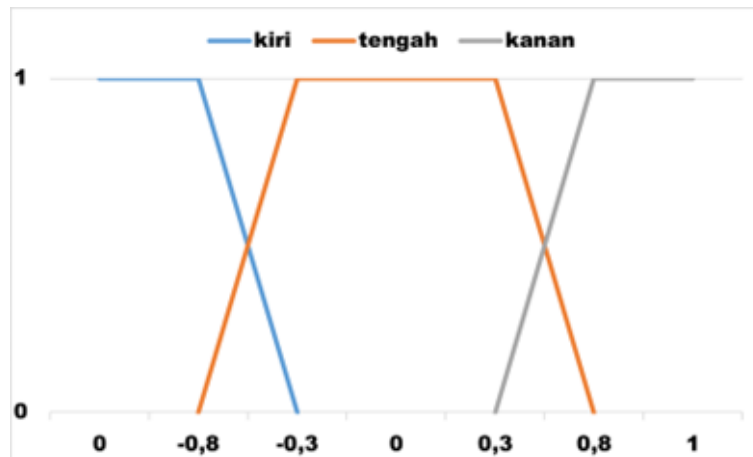
$$\begin{aligned} \mu_c &= (\text{Pointer} - a) / (b - a) \\ &= (\text{Pointer} - (-0.8)) / ((-0.3) - (-0.8)) \\ &= (\text{Pointer} - (-0.8)) / (-0.5) \\ &= 1.5\text{TA} - 1.6 \end{aligned}$$

Tabel 2: Membership Input *Fuzzy* Horizontal

Membership	Persamaan	Rentang Nilai	Kategori
μ_a	1	$<(-0.8)$	Kiri
μ_b	$0.6 - 1.5$ Pointer	$(-0.8 \leq \text{Pointer} \leq (-0.3))$	Kiri
μ_c	$1.5\text{TA} - 1.6$	$(-0.8) \leq \text{Pointer} \leq (-0.3)$	Tengah
μ_d	1	$(-0.3) \leq \text{Pointer} \leq (0.3)$	Tengah
μ_e	$0.6 - 1.5$ Pointer	$(0.3) \leq \text{Pointer} \leq (0.8)$	Tengah

Membership	Persamaan	Rentang Nilai	Kategori
μ_f	1.5TA-1.6	$(0.3) \leq \text{Pointer} \leq (0.8)$	Kanan
μ_g	1	>0.8	Kanan

2) **Input fuzzy Vertikal**



Gambar 14. Himpunan Fuzzy Vertikal

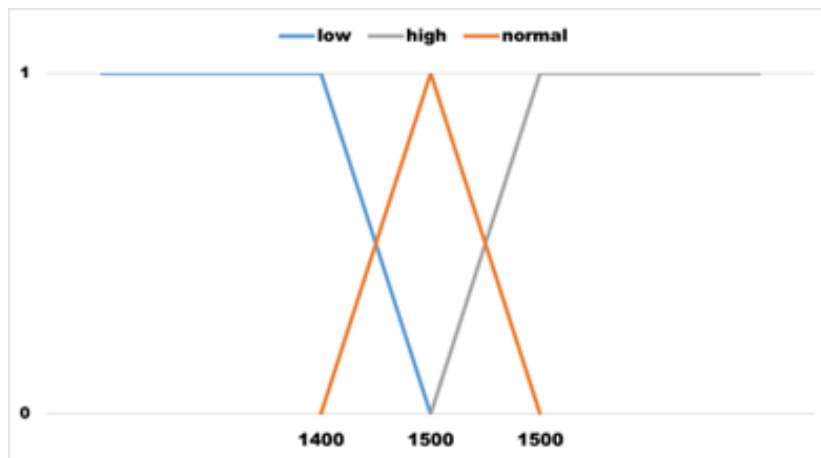
Tabel 3: Input Fuzzy Vertikal

Fungsi	Nama Variablel	Keanggotaan
Y Axis	Atas	<-0.8 -0.8 -0.3
	Tengah	-0.8 0 0.8
	Bawah	0.3 0.8 >0.8

Tabel 4: Nilai Membership Input Fuzzy Vertikal

Membership	Persamaan	Rentang Nilai	Kategori
μ_a	1	$<(-0.8)$	Atas
μ_b	0.6-1.5 Pointer	$(-0.8) \leq \text{Pointer} \leq (-0.3)$	Atas
μ_c	1.5TA-1.6	$(-0.8) \leq \text{Pointer} \leq (-0.3)$	Tengah
μ_d	1	$(-0.3) \leq \text{Pointer} \leq (0.3)$	Tengah
μ_e	0.6-1.5 Pointer	$(0.3) \leq \text{Pointer} \leq (0.8)$	Tengah
μ_f	1.5TA-1.6	$(0.3) \leq \text{Pointer} \leq (0.8)$	Bawah
μ_g	1	>0.8	Bawah

Penentuan Output *Fuzzy Logic*



Gambar 15. Himpunan Output *Fuzzy*

Tabel 5: Output *Fuzzy Logic*

Fungsi	Nama Variablel	Keanggotaan
Output Motor	Rendah	[0 1400 1500]
	Normal	[1400 1500 1600]
	Tinggi	[1500 1600 1300]

Tabel 6: Nilai Membership Output *Fuzzy Logic*

Membership	Persamaan	Rentang Nilai	Kategori
μ_a	1	$x < 1400$	Rendah
μ_b	$7 - 0.04x$	$1400 \leq x \leq 1500$	Rendah
μ_c	$0.04x - 6$	$1400 \leq x \leq 1500$	Sedang
μ_d	$8 - 0.04x$	$1500 \leq x \leq 1600$	Sedang
μ_e	$0.04x - 7$	$1500 \leq x \leq 1600$	Tinggi
μ_f	1	$x > 1600$	Tinggi

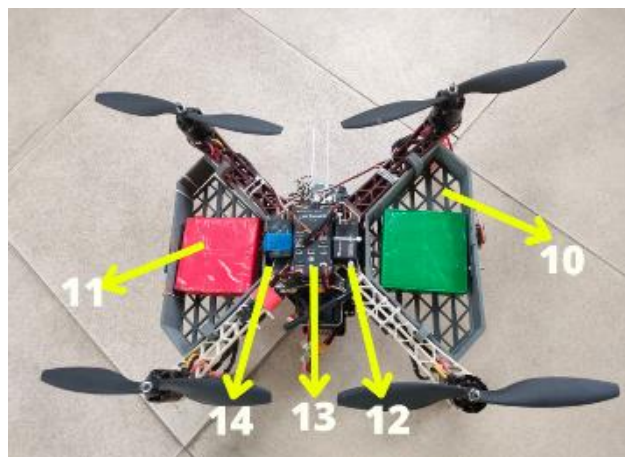
HASIL DAN PEMBAHASAN
Hasil Perencanaan Hardware



Gambar 16. Tampak Depan



Gambar 17. Tampak Samping



Gambar 18. Tampak Atas

Hasil Perencanaan Software

```
paksi@PaksiPratama:~$ python3 main.py
paksi@PaksiPratama:~/d$ python3 main.py
Jarak: 0.98 cm (0.0)
Jarak: 0.98 cm (-14.78)
Jarak: 0.98 cm (-29.0)
Jarak: 0.98 cm (-43.0)
Jarak: 0.98 cm (-48.5)
Jarak: 0.98 cm (-41.42)
Jarak: 0.98 cm (0)
Jarak: 0.98 cm (39.0)
Jarak: 0.98 cm (57.0)
Jarak: 0.98 cm (64.0)
Jarak: 0.98 cm (60.0)
Jarak: 0.98 cm (69.0)
Jarak: 0.98 cm (79.0)
Jarak: 0.98 cm (64.21)
Jarak: 0.98 cm (0)
Jarak: 0.98 cm (50.66)
Jarak: 0.98 cm (44.25)
Jarak: 0.98 cm (0)
Jarak: 0.98 cm (0.0)
Jarak: 0.98 cm (32.5)
Jarak: 0.98 cm (12.0)
Jarak: 0.98 cm (31.0)
Jarak: 0.98 cm (0.0)
```

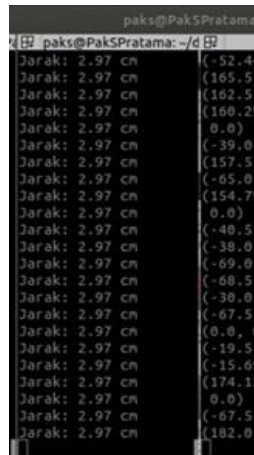
Gambar 19. LiDAR 1 meter

```
paksi@PaksiPratama:~$ python3 main.py
paksi@PaksiPratama:~/d$ python3 main.py
Jarak: 1.51 cm (-24.0)
Jarak: 1.51 cm (-14.0)
Jarak: 1.51 cm (-21.66)
Jarak: 1.51 cm (0)
Jarak: 1.51 cm (-14.5)
Jarak: 1.51 cm (0.0)
Jarak: 1.51 cm (-22.5)
Jarak: 1.51 cm (-12.5)
Jarak: 1.51 cm (-11.0)
Jarak: 1.51 cm (-19.5)
Jarak: 1.51 cm (-11.0)
Jarak: 1.51 cm (0.0)
Jarak: 1.51 cm (-10.5)
Jarak: 1.51 cm (-7.5)
Jarak: 1.51 cm (-9.10)
Jarak: 1.51 cm (0.0)
Jarak: 1.51 cm (-6.0)
Jarak: 1.51 cm (0.0)
Jarak: 1.51 cm (-8.71)
Jarak: 1.51 cm (0.0)
Jarak: 1.51 cm (4.5)
Jarak: 1.51 cm (8.7348)
```

Gambar 20. LiDAR 1.5 meter

```
paksi@PaksiPratama:~$ python3 main.py
paksi@PaksiPratama:~/d$ python3 main.py
Jarak: 1.98 cm (12.51)
Jarak: 1.98 cm (0.0)
Jarak: 1.98 cm (8.5)
Jarak: 1.98 cm (11.5)
Jarak: 1.98 cm (12.0)
Jarak: 1.98 cm (15.5)
Jarak: 1.98 cm (21.0)
Jarak: 1.98 cm (17.5)
Jarak: 1.98 cm (0.0)
Jarak: 1.98 cm (0.0)
Jarak: 1.98 cm (0.0)
Jarak: 1.98 cm (12.757)
Jarak: 1.98 cm (0.0)
Jarak: 1.98 cm (18.5)
Jarak: 1.98 cm (18.0)
Jarak: 1.98 cm (19.5)
Jarak: 1.98 cm (21.5)
Jarak: 1.98 cm (18.5)
Jarak: 1.98 cm (25.5)
Jarak: 1.98 cm (16.758)
Jarak: 1.98 cm (0.0)
Jarak: 1.98 cm (10.5)
Jarak: 1.98 cm (1.0)
```

Gambar 21. LiDAR 2 meter



Gambar 22. LiDAR 3 meter

Pengujian akurasi sensor jarak yaitu lidar dilakukan dengan tujuan untuk mengetahui tingkat akurasi sensor lidar dalam pengukuran jarak pada Quadcopter. Pengujian sensor LiDAR diatas dilakukan ketika Quadcopter memulai take-off dan kemudian mempertahankan posisi diudara (hover) sesuai dari perintah ketinggian yang sudah disetting sebelumnya.

Hasil Pengujian Lidar

Tabel 7. Pengujian LiDAR Dengan Jarak Sebenarnya

No	Jarak Sebenarnya (cm)	Lidar (cm)	Selisih	Error (%)
1	5	0	5	100
2	10	5	5	50
3	50	48	2	4
4	100	98	2	2
5	150	151	1	0.66
6	200	198	2	1
7	250	250	0	0
8	300	297	3	1
9	350	352	2	0.57
10	400	401	1	0.25
Rata – rata error				15.94 %

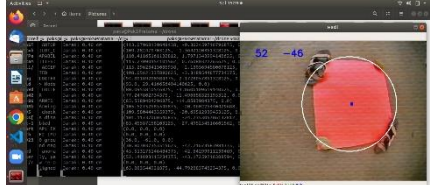
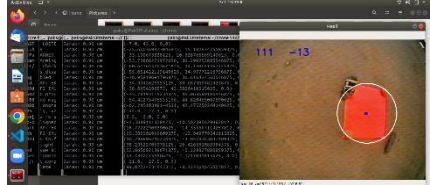
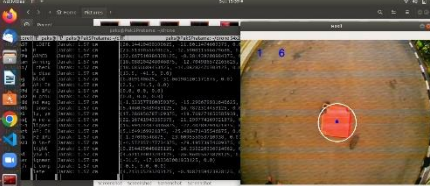
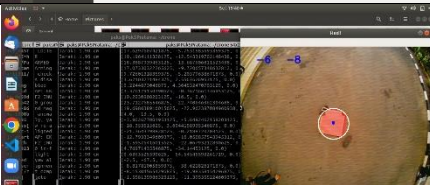
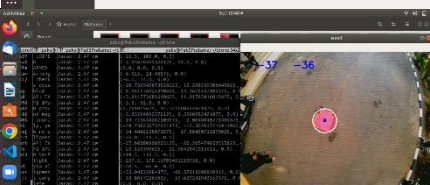
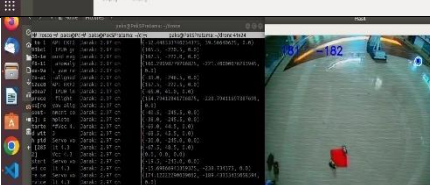
$$\text{Persentase Error} = \frac{(\text{Nilai yang diamati} - \text{nilai sebenarnya})}{\text{Nilai sebenarnya}} \times 100\%$$

$$\text{Contoh pada no tabel no. 2} = \frac{\text{Selisih}}{10} = \frac{5}{10} \times 100\% = 50\%$$

- Maka Rata – rata persentase error dari keseluruhan data = $\frac{159.48}{10} = 15,94\%$
- Persentase Akurasi = $100\% - \text{Rata-rata persentase error}$
- = $100\% - 15,94\%$
- = $84,06\%$

Hasil Pengujian Blob Detection

Tabel 8: Hasil Pengujian Blob Detection

No	Ketinggian (m)	Hasil Deteksi Objek	Deteksi	Status
1	0.5		Terdeteksi	Dropping
2	1		Terdeteksi	Dropping
3	1.5		Terdeteksi	Dropping
4	2		Terdeteksi	Dropping
5	2.5		Terdeteksi	Dropping
6	3		Tidak Terdeteksi	Tidak Dropping

Hasil deteksi objek dari ketinggian tertentu menggunakan kamera analog FPV dan mendapatkan hasil yang cukup baik, meskipun berada pada kondisi pencahayaan yang berbeda. Pengujian alat ini dilaksanakan saat pagi hari berkisar pukul 06.00–8.00 WIB dikarenakan menghindari adanya kondisi angin yang cukup kencang. Dan dapat dilihat pada hasil tabel objek terdeteksi sampai dengan jarak 3 meter, dan untuk jarak 4 meter ke atas objek sudah tidak dapat terdeteksi dikarenakan pengaruh pencahayaan dan kamera yang digunakan. Maka dari itu Quadcopter dapat melakukan dropping kit pada saat kondisi “Terdeteksi” yaitu pada ketinggian kurang lebih 1-3 meter.

Menurut Haniffullah, Bangsa, and Stefanie (2020) quadcopter dapat mempertahankan posisinya dengan memanfaatkan data valid GPS dan accelerometer. Data tersebut dibandingkan dengan nilai aktual posisi quadcopter, kemudian dilakukan koreksi hingga nilai aktualnya sesuai atau mendekati dengan nilai set point yang dibaca oleh GPS. Dari hasil pengujian yang didapat, Drone skyscraper mampu mempertahankan posisinya pada saat arm robot yang dibawanya sedang melakukan pembersihan kaca jendela. Sistem kendali position hold telah terealisasi dengan persentase error untuk latitude 0,015 % dan untuk longitude 0,55 %.

SIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini yaitu, penelitian ini sudah berhasil mengimplementasikan metode blob detection untuk melakukan misi dropping kit pada Quadcopter, hal ini dapat dilihat dari hasil perangkat lunak yang mendeteksi objek yaitu warna merah dan menghasilkan koordinat titik tengah pada objek warna tersebut, Nilai yang dihasilkan tersebut yaitu menggunakan nilai minimum dimana (0, 165, 206) dan nilai maksimum (255, 255, 255). Penerapan blob detection tersebut dengan menggunakan kamera FPV (*First Person View*) Caddx Ratel. Pengujian Quadcopter ini menggunakan sensor LiDAR Benewake untuk mendeteksi dan mengukur jarak objek dengan presisi yang lumayan tinggi. Percobaan yang telah dilakukan dalam melakukan misi dropping kit ini yaitu mulai dari ketinggian 0.5 meter sampai dengan 3 meter. Dan dari hasil percobaan tersebut menyatakan bahwa mulai dari ketinggian ½ meter sampai dengan 2 1/2 meter Quadcopter mampu mendeteksi target warna dan pada ketinggian 3 meter Quadcopter tidak mampu mendeteksi target warna. Maka dari itu blob detection dapat dicapai sampai dengan ketinggian ± 2.5 meter dari permukaan Quadcopter melakukan take-off.

DAFTAR PUSTAKA

- Alamsyah, Sayyidul Aulia, and Muhammad Rivai. 2020. "Implementasi Lidar Sebagai Kontrol Ketinggian Quadcopter." *Jurnal Teknik ITS* 8(2). doi: 10.12962/j23373539.v8i2.43034.
- Aziz, Fardiansyah Nur, and Masduki Zakarijah. 2022. "TF-Mini LiDAR Sensor Performance Analysis for Distance Measurement." *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi* 11(3):192–98.
- Hadi, Ghozali S., Rivaldy Varianto, Bambang Riyanto T, and Agus Budiyo. 2014. "Autonomous UAV System Development for Payload Dropping Mission." (March 2015).
- Haniffullah, Al, Insani Abdi Bangsa, and Arnisa Stefanie. 2020. "Implementasi Quadcopter Pembawa Arm Robot Pembersih Kaca Jendela Gedung Bertingkat." *Edu ElektriKa Journal* 9(2):36–42.
- Hochstenbach, Menno, Cyriel Notteboom, Bart Theys, and Joris De Schutter. 2015. "Design and Control of an Unmanned Aerial Vehicle for Autonomous Parcel Delivery with Transition from Vertical Take-off to Forward Flight - VertiKUL, a Quadcopter Tailsitter." *International Journal of Micro Air Vehicles* 7(4):395–405. doi: 10.1260/1756-8293.7.4.395.
- Manan, Abdul, Agung Nugroho Jati S. T, and M. T. Randy Erfa Saputra S. T. 2020. "Perancangan Dan Simulasi Kendali Otonom Untuk Take-Off Dan Landing Pada Quadcopter Drone Design and Simulation Autonomy Control for Take-Off and Landing on Quadcopter Drone." 7(2):4864–71.
- Palaha, Fadhli, Jamar Teles, and Yolnasdi Yolnasdi. 2020. "Analisa Rancangan Keseimbangan Menggunakan Sensor Imu Type-Mpu6050 Pada Quadcopter." *Sainstek (e-Journal)* 8(2):96–104.
- Pamungkas, Elki Muhamad, Bakhtiar Alldino Ardi Sumbodo, and Ika Candradewi. 2017. "Sistem Pendeteksi Dan Pelacakan Bola Dengan Metode Hough Circle Transform, Blob Detection, Dan Camshift Menggunakan AR.Drone." *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)* 7(1):1. doi: 10.22146/ijeis.15405.
- Ramadhan, Rofy Wahyu, Ahmad Wilda Yulianto, and Yoyok Heru Prasetyo Isnomo. 2022. "Quadcopter Take Off and Landing System with Blob Detection Method and Optical Flow." *Jurnal Jartel Jurnal Jaringan Telekomunikasi* 1–5.
- Yang, Kang, Guang You Yang, and S. Isi Huang Fu. 2020. "Research of Control System for Plant Protection UAV Based on Pixhawk." *Procedia Computer Science* 166:371–75. doi: 10.1016/j.procs.2020.02.082.



© 2022 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY SA) license (<https://creativecommons.org/licenses/by-sa/4.0/>).