

SISTEM OBJECT TRACKING PADA QUADCOPTER MENGGUNAKAN SEGMENTASI CITRA DENGAN DETEksi WARNA HSV DAN METODE REGRESI LINIER BERBASIS RASPBERRY PI

Fahmi Erza^{*1}, Hurriyatul Fitriyah², Eko Setiawan³

^{1,2,3}Universitas Brawijaya, Malang
Email: ¹fahmierza7@gmail.com, ²hfifitriyah@ub.ac.id, ³ekosetiawan@ub.ac.id
^{*}Penulis Korespondensi

(Naskah masuk: 16 Desember 2022, diterima untuk diterbitkan: 27 Desember 2022)

Abstrak

Saat ini, banyak aplikasi perangkat cerdas yang dikembangkan untuk melakukan tugas secara mandiri tanpa menerima perintah dari manusia. Oleh karena itu, mengembangkan sistem yang memungkinkan perangkat untuk melakukan tugas pengawasan seperti mendeteksi dan melacak objek bergerak akan memungkinkan tugas yang lebih canggih untuk diterapkan pada robot di masa depan. Teknologi *Quadcopter* sesungguhnya dapat memudahkan pekerjaan manusia dalam melakukan pengawasan dan pelacakan seperti pada kasus pelacakan lansia atau ABK (Anak Berkebutuhan Khusus) secara otomatis agar kerabat dapat melakukan pengawasan dengan menggunakan drone. Sehingga penelitian ini dilakukan untuk membuat sebuah sistem pada drone atau *quadcopter* agar dapat mendeteksi objek dan mengikutinya. Pada implementasinya, orang yang berkebutuhan khusus dan membutuhkan pengawasan akan mengenakan atribut berupa topi dengan warna solid. Warna topi tersebut akan dijadikan acuan untuk *threshold* segmentasi warna untuk mendeteksi objek topi tersebut dengan pemrosesan citra digital. Pergerakan drone ditentukan oleh prediksi jarak, sudut, dan ketinggian objek berdasarkan regresi linier yang dihasilkan dari 123 data latih. Hasil deteksi sistem juga cukup sesuai dengan pergerakan drone ketika diuji dengan 27 data. Akurasi dari prediksi gerak *pitch* adalah 84%, prediksi gerak *yaw* adalah 94%, dan prediksi gerak *up/down* adalah 91,5%. Adapun waktu komputasinya adalah 0.175829662 detik per frame.

Kata kunci: Drone, Deteksi Objek, Segmentasi Warna, Raspberry pi, Lansia, ABK.

OBJECT TRACKING SYSTEM ON QUADCOPTER USING IMAGE SEGMENTATION WITH HSV COLOR DETECTION AND LINEAR REGRESSION METHOD BASED ON RASPBERRY PI

Abstract

Nowadays, many intelligent device applications are developed to perform tasks independently without receiving commands from humans. Therefore, developing systems that allow devices to perform surveillance tasks such as detecting and tracking moving objects will allow more sophisticated tasks to be applied to robots in the future. Quadcopter technology can actually facilitate human work in monitoring and tracking, such as in the case of tracking the elderly or children with special needs automatically so that relatives can carry out surveillance using drones. So this research was planned to create a system on a drone so it can detect objects and follow them. In its implementation, people with special needs and need supervision will wear an attribute in the form of a hat with a solid color. The color of the hat will be used as references for the color segmentation threshold to detect the hat object with digital image processing. The movement of the drone is determined by the prediction of the distance, angle, and height of the object based on linear regression generated from 123 training data. The system detection results are also quite in accordance with the movement of the drone when tested with 27 data. The accuracy of pitch motion prediction is 84%, yaw motion prediction is 94%, and up/down motion prediction is 91.5%. The computation time is 0.175829662 seconds per frame.

Keywords: Drone, Object tracking, Color Segmentation, Raspberry pi, Elderly, a special needs children.

1. PENDAHULUAN

Saat ini, banyak aplikasi perangkat cerdas yang dikembangkan untuk melakukan tugas secara mandiri tanpa menerima perintah dari manusia. Oleh karena itu, mengembangkan sistem yang memungkinkan perangkat untuk melakukan tugas pengawasan seperti

mendeteksi dan melacak objek bergerak akan memungkinkan tugas yang lebih canggih untuk diterapkan pada robot di masa depan.

Di sekitar kita terdapat beberapa kalangan yang membutuhkan perhatian lebih serta pengawasan seperti kalangan Lansia (Lanjut Usia) dan ABK (Anak Berkebutuhan Khusus). Kalangan Lansia dan

ABK membutuhkan pengawasan untuk dapat dilakukan reaksi yang efektif ketika terjadi hal-hal yang tidak diinginkan. Menurut Badan Pusat Statistik (BPS), jumlah Lansia atau orang dengan usia di atas 65 tahun di Indonesia pada tahun 2021 adalah sebanyak ± 16 juta jiwa atau 5,95% dari total penduduk Indonesia yang tercatat sebanyak 270,2 juta jiwa dan jumlah ABK di Indonesia terhitung pada 2017 adalah sebanyak ± 1,6 juta jiwa.

Di era modern ini, sudah banyak teknologi robotika yang mampu bergerak untuk memenuhi kebutuhan atau membantu pekerjaan manusia. Salah satunya adalah *quadcopter* atau biasa disebut drone. *Quadcopter* adalah benda terbang tanpa awak yang memiliki 4 baling-baling dan biasanya digunakan untuk mengambil gambar atau video dari sudut yang tinggi. Fitur *quadcopter* ini sesungguhnya dapat disandingkan dengan komputasi citra digital sehingga dapat memberikan berbagai solusi dan manfaat seperti untuk mendeteksi dan menganalisis suatu objek, kejadian, atau bahkan bencana. Teknologi *quadcopter* sesungguhnya dapat memudahkan pekerjaan manusia dalam melakukan pengawasan dan pelacakan seperti pada kasus pelacakan lansia atau ABK secara otomatis agar kerabat dapat melakukan pengawasan dengan menggunakan drone.

Penelitian dan implementasi *quadcopter* dengan memanfaatkan komputasi citra untuk mendeteksi manusia sebenarnya sudah dilakukan oleh beberapa peneliti seperti yang dilakukan oleh Rana Praful George dan V. Prakash pada tahun 2018 tentang bagaimana *quadcopter* dapat mendeteksi manusia dengan komputasi citra digital. Pada penelitian tersebut *quadcopter* bergerak sesuai perintah atau rute, lalu ketika dideteksi terdapat objek berupa manusia, perangkat akan mengirimkan informasi menuju pusat atau server. Oleh karena itu, peneliti berusaha mengembangkan suatu desain sistem yang membuat *quadcopter* dapat bergerak mengikuti objek manusia yang ditargetkan untuk pengawasan dan pelacakan.

Sehingga dengan dirancangnya sistem *object tracking* pada *quadcopter* menggunakan segmentasi citra dengan deteksi warna HSV dan metode regresi linier berbasis *raspberry pi* ini diharapkan dapat membantu manusia dalam pengawasan dan pelacakan secara otomatis. Penelitian ini akan menggunakan drone DJI Tello untuk melakukan pelacakan objek. DJI Tello terhubung dengan mikro komputer *raspberry pi* secara nirkabel. Citra yang diterima dari DJI Tello akan diproses oleh *raspberry pi* dengan melakukan segmentasi citra untuk mendeteksi objek dan menentukan nilai kontrol yang akan dikirimkan kepada DJI Tello sehingga dapat melakukan pelacakan objek. Metode regresi linier digunakan untuk menghitung prediksi nilai kontrol yang digunakan DJI Tello untuk dapat bergerak mengikuti objek. Adapun tantangan pada penelitian ini seperti bagaimana sistem bisa mendeteksi objek dengan berbagai posenya, permasalahan pada video

yang diproses secara *real-time*, dan kesesuaian hasil keluaran yang diimplementasikan untuk mengontrol pergerakan *quadcopter* berdasarkan pergerakan objek serta kecepatan waktu komputasinya.

2. DASAR TEORI

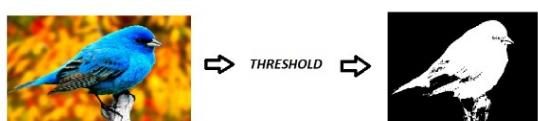
2.1. Citra Digital

Pada dasarnya citra digital berasal dari pantulan cahaya pada objek tertentu yang ditangkap atau direkam menggunakan kamera. Citra digital juga bisa terbuat dari rekayasa atau terbentuknya kumpulan titik atau piksel yang membentuk citra pada komputer.

2.2. Segmentasi Citra

Segmentasi HSV merupakan proses pemisahan objek dengan seleksi warna berdasarkan nilai *Hue*, *Saturation*, dan *Value*. *Hue* merupakan atribut yang merepresentasikan warna murni. *Saturation* merupakan atribut yang menunjukkan efek cahaya putih yang mempengaruhi tingkat dominasi warna. Dan *Value* merupakan atribut yang menunjukkan perbedaan kecerahan pada warna murni.

Pada metode segmentasi dengan deteksi warna HSV menurut Giannakopoulos (2008), dilakukan pemilihan sampel piksel sebagai acuan warna untuk membentuk segmen yang diinginkan. Untuk membentuk segmen sesuai dengan warna yang diinginkan maka ditentukan nilai toleransi pada setiap dimensi warna HSV, kemudian nilai toleransi tersebut digunakan dalam perhitungan proses adaptive *threshold*. Hasil dari proses *threshold* tersebut akan membentuk segmen area dengan warna sesuai toleransi yang diinginkan seperti ditampilkan pada gambar 1 berikut:



Gambar 1 Segmentasi Citra

2.3. *Quadcopter*

Quadcopter adalah perangkat yang memiliki 4 buah rotor atau baling-baling sebagai penggerak yang menghasilkan daya angkat. *Quadcopter* dapat melakukan *take off* dan *landing* secara vertikal sehingga dapat bermanfaat untuk memulai penerbangan dan pendaratan di tempat yang sempit. Pergerakan pada *quadcopter* lebih dikenal dengan istilah *pitch* (bergerak maju atau mundur), *roll* (bergerak kiri atau kanan), dan *yaw* (rotasi kiri atau rotasi kanan). *Quadcopter* yang digunakan pada penelitian ini adalah DJI Tello.

2.4. *Raspberry pi*

Raspberry pi merupakan komputer kecil yang biasa digunakan untuk suatu tujuan pengolahan

tertentu. Perangkat ini merupakan salah satu kontroler yang mampu mengolah data masukan berupa citra atau video. *Raspberry pi* dikembangkan oleh yayasan nirlaba, Rasberry Pi Foundation, yang digawangi sejumlah pengembang dan ahli komputer dari Universitas Cambridge, Inggris.

Pada penelitian ini dimana citra merupakan sebuah masukan sistem, Raspi digunakan untuk mengolah citra digital mulai dari akuisisi data, pra-pengolahan, hingga pengolahan deteksi target dan mengirimkan perintah kepada *quadcopter*. Sehingga *quadcopter* dapat bergerak sesuai pergerakan target yang diikuti.

2.5. Python

Python adalah bahasa pemrograman yang berorientasi objek dinamis dan hal itu menyebabkan penggunaannya sering menggunakan *python* untuk pengembangan perangkat lunak dan komputasi sains tingkat tinggi dikarenakan penggunaannya sederhana dan mudah dimengerti.

2.6. OpenCV

OpenCV atau *Open source Computer Vision* merupakan sebuah pustaka perangkat lunak yang digunakan untuk pengolahan citra dinamis secara *real-time*. Pustaka ini berada pada lisensi BSD yang menunjukkan jika pustaka ini bersifat *open source*. Pustaka *opencv* digunakan pada penelitian ini untuk pemanggilan metode pra-pengolahan hingga pendekripsi objek pada citra yang berasal dari drone DJI Tello.

2.7. Regresi Linier

Regresi linier adalah salah satu metode pengenalan pola untuk memprediksi suatu nilai berdasarkan masukan pada fungsi linier. Terdapat dua variabel penting yaitu variabel *X* sebagai masukan dan bersifat independen, dan variabel *Y* sebagai hasil yang bersifat dependen. Dalam pembuatan fungsi regresi linier diperlukan beberapa data latih yang sudah memiliki variabel *X* dan *Y* pada setiap sampel. Perhitungan regresi linier akan menghasilkan sebuah fungsi $aX + b = Y$, dimana pada percobaan penggunaannya data masukan ada pada variabel *X* dan akan menghasilkan hasil prediksi pada variabel *Y*. Adapun rumus penentuan persamaan regresi linear ada pada persamaan 2.1.

$$a = \frac{n(\sum XY) - (\sum X)(\sum Y)}{n(\sum x^2) - (\sum X)^2} \quad b = \frac{\sum Y - a(\sum X)}{n} \quad (2.1)$$

Keterangan:

- x* : Variabel independen
- y* : Variabel dependen
- n* : Jumlah sampel
- a* : Koefisien regresi
- b* : Konstanta

2.8. Mean absolute percentage error

Pada sebuah penelitian yang bersifat kuantitatif, salah satu hasil penelitiannya adalah akurasi dan error sistem. Hal ini dibutuhkan untuk menentukan seberapa bisa diandalkannya sebuah sistem yang dibuat atau diteliti. Ada banyak metode untuk menghitung error dan akurasi. Namun penelitian ini menggunakan metode *Mean absolute percentage error* (MAPE) untuk mencari akurasi dari percobaan sistem. MAPE adalah metode perhitungan untuk mencari persentasi error pada percobaan. Rumus perhitungan MAPE ada pada persamaan 2.2.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{A_i - F_i}{A_i} \right| \times 100\% \quad (2.2)$$

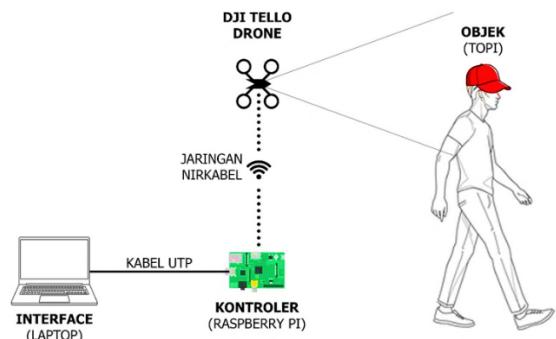
Keterangan:

- n* : Ukuran sampel
- A_i* : Nilai data aktual
- F_i* : Nilai data peramalan

3. PERANCANGAN DAN IMPLEMENTASI

3.1. Perancangan Perangkat Keras

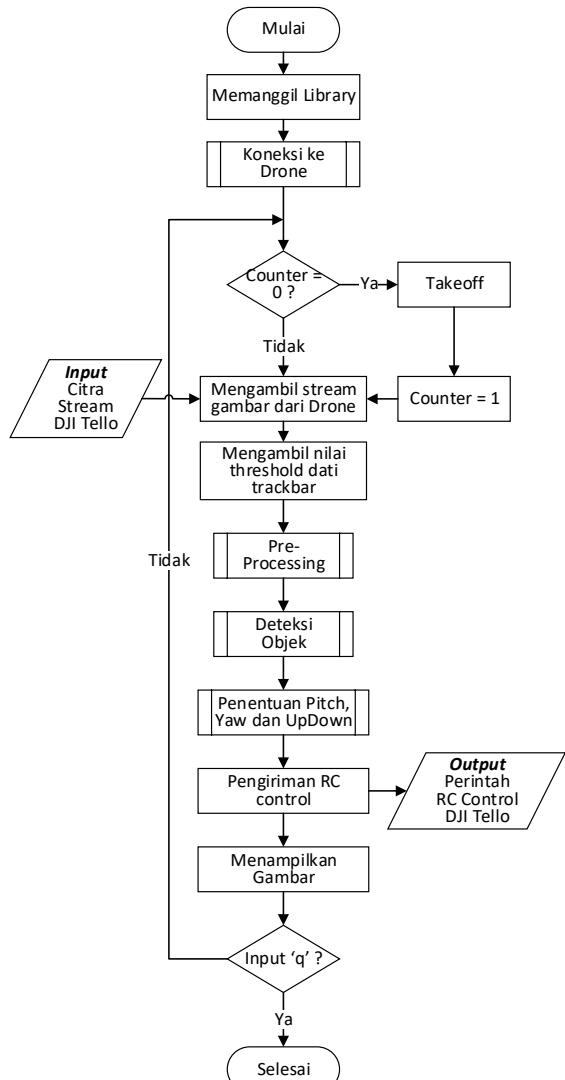
Perancangan alat ini terdiri dari 2 bagian yaitu Kontroler *Raspberry pi* dan DJI Tello Drone. Drone akan memancarkan *hotspot* yang terbuka dan untuk siapa pun, lalu kontroler *Raspberry pi* akan terhubung dengan *hotspot* tersebut untuk dapat mengirim atau menerima data dari DJI Tello Drone melalui *WiFi*. sehingga drone dapat bergerak mengikuti objek. Sistem ini juga menggunakan laptop yang berfungsi untuk menampilkan antarmuka dalam melakukan percobaan dan pengujian. Sistem bisa berjalan walaupun tidak menggunakan antarmuka (laptop). Ilustrasi perancangan perangkat keras ditampilkan pada gambar 2.



Gambar 2 Ilustrasi Perancangan Perangkat Keras

3.2. Perancangan Perangkat Lunak

Perancangan program ini meliputi keseluruhan proses dan sub-program yang terdapat pada perancangan perangkat lunak mulai dari masukan citra, proses pendekripsi objek dengan segmentasi, hingga proses penentuan gerak dengan regresi linier dan pengiriman *output* berupa nilai kontrol drone. Perancangan program keseluruhan diterangkan pada gambar 3.

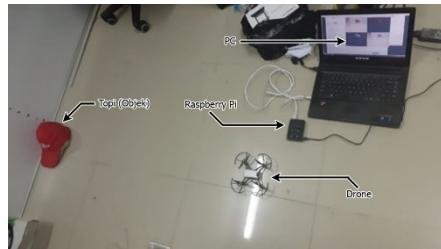


Gambar 3 Diagram Alir Perangkat Lunak

Gambar 3 menampilkan diagram yang mengalirkan proses – proses yang dilakukan dari awal hingga akhir sistem. Pada sistem terdapat pengecekan kondisi variabel *counter* untuk melakukan *take off* di awal menjalankan program. Di dalam program masih terdapat beberapa sub-program yang akan mengolah data pada sistem. Adapun sub-programnya antara lain adalah: pengkoneksi ke drone, pre-processing, deteksi objek, serta penentuan RC *control*. Setelah menjalankan beberapa sub-program tersebut, data perintah RC akan dikirimkan ke drone agar drone dapat bergerak mengikuti objek.

3.3. Implementasi Perangkat Keras

Perangkat keras adalah komponen-komponen fisik yang terdapat di dalam sistem yang dirancang ini. Pada penelitian sistem human follower pada *quadcopter* dengan metode segmentasi citra dengan deteksi warna HSV ini, perangkat keras yang digunakan adalah Personal komputer, *Raspberry pi*, dan juga DJI Tello Drone.



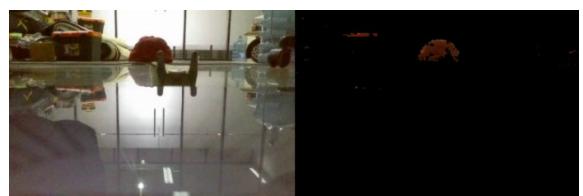
Gambar 4 Implementasi Perangkat Keras.

Gambar 4 menampilkan bagaimana implementasi perangkat keras pada sistem. Personal komputer akan terhubung dengan *Raspberry pi* untuk dapat mengoperasikan *raspberry pi*. Hubungan dapat dilakukan melalui nirkabel atau pun dengan kabel UTP (*Unshielded Twisted Pair*). Ketika sudah terhubung dengan media jaringan. Pengguna akan memilih atau memasukan alamat host dari *raspberry pi*. Lalu tampilan GUI *raspberry pi* akan dapat dioperasikan di personal komputer.

Melalui tampilan GUI pada persoal komputer, Pengguna bisa menghubungkan *raspberry pi* dengan drone melalui jaringan *Wifi*. Dan juga harus dipastikan client dari *hotspot* DJI tello drone hanyalah pengguna saja agar tidak terjadi gangguan dalam pengiriman perintah atau penerimaan data. Ketika *raspberry pi* dan drone sudah terhubung melalui *Wifi*, barulah *raspberry pi* dapat memerikan perintah atau dapat pula menerima data dari drone.

3.4. Implementasi Perangkat Lunak

Implementasi perangkat lunak pada penelitian ini membahas bagaimana sebuah program dapat menerima citra dari drone dan melakukan deteksi objek hingga mengirimkan perintah kepada drone.



Gambar 5 Masking Object

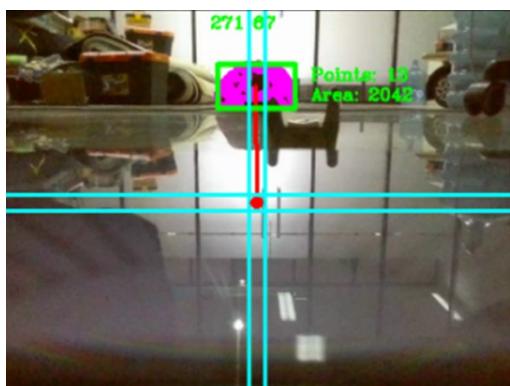
Terlihat pada gambar 5 di atas bagaimana perubahan citra yang didapat dari kamera drone ketika melakukan *masking*. Gambar 3.4 bagian kiri adalah masukan gambar yang belum diolah. Gambar 3.4 bagian kanan adalah hasil pengolahan citra berupa *masking* yang hanya meloskan warna merah saja.

Setelah segmentasi citra dengan *masking* objek, Langkah yang dilakukan adalah mendeteksi tepi dengan fungsi *canny*. *Canny* adalah fungsi yang terdapat pada *OpenCV* untuk menemukan tepi pada objek yang telah disegmentasi.



Gambar 6 Deteksi Tepi

Gambar 6 menunjukkan perubahan citra setelah melakukan pendeteksian tepi objek hasil *masking*. Gambar 6 bagian atas adalah masukan gambar yang belum diolah. Gambar 6 bagian bawah adalah hasil pengolahan citra berupa deteksi tepi yang akan mengelilingi *masking* sehingga membentuk objek.



Gambar 7 Bounding rectangle

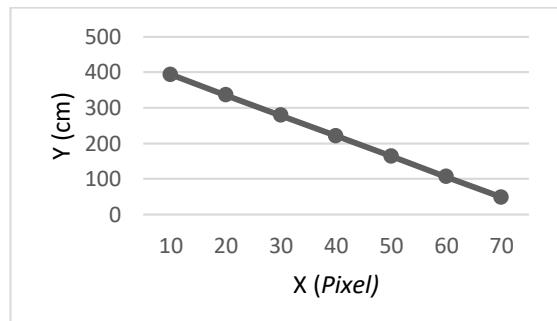
Gambar 7 adalah hasil akhir dari pendeteksian objek yaitu membuat *bounding rectangle*. *Bounding rectangle* ditandai dengan sebuah segi empat dengan garis berwarna hijau. *Bounding rectangle* didapatkan dari berkumpulnya garis *contour* yang melebihi batas area (garis berwarna merah muda pada gambar). *Bounding rectangle* juga menghasilkan empat data penting dalam perhitungan sistem yaitu nilai *x,y,w,h*. Nilai *x* dan *y* adalah titik koordinat pada salah satu sudut *bounding rectangle*. Nilai *w* atau *width* adalah jarak piksel dari titik koordinat ke arah horizontal. Sedangkan nilai *h* atau *height* adalah jarak piksel dari titik koordinat ke arah vertikal. Nilai-nilai tersebut yang akan digunakan dalam perhitungan regresi linier dalam penentuan gerak drone.

3.5. Implementasi Pengambilan Data

Pengambilan dataset dilakukan untuk menemukan fungsi regresi linear yang akan digunakan untuk menentukan masukan pergerakan drone seperti *pitch*, *yaw*, *up* dan *down*. Pada penelitian ini, dataset diperoleh dengan cara memberikan objek tertangkap kamera drone dengan posisi yang diketahui dan menentukan nilai thrust yang harus dilakukan drone untuk membuat objek berada di tengah gambar.

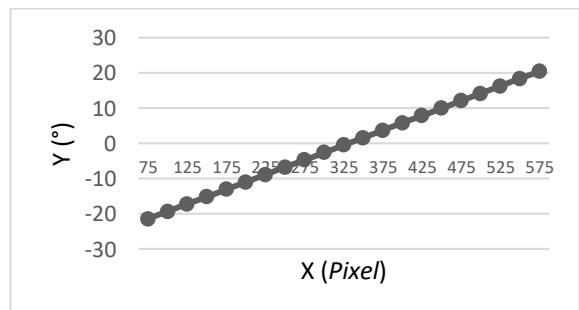
3.5.1. Pengambilan Data Latih *Pitch*

Pengambilan data latih ini bertujuan untuk membuat fungsi regresi linier dalam penentuan gerak maju dan mundur drone atau biasa disebut *pitch*. Jarak normal antara objek dan drone yang ditentukan adalah 1 meter – 1,5 meter. Masukan fungsinya adalah nilai *height* dari *bounding rectangle* dan keluarannya berupa nilai gerak *pitch*. Fungsi regresi linear berdasarkan 41 data latih *pitch* adalah $y = -5.747192413(x) + 450.72573$ dan menghasilkan grafik fungsi seperti pada gambar 8.

Gambar 8 Grafik Fungsi Regresi Linier *Pitch*

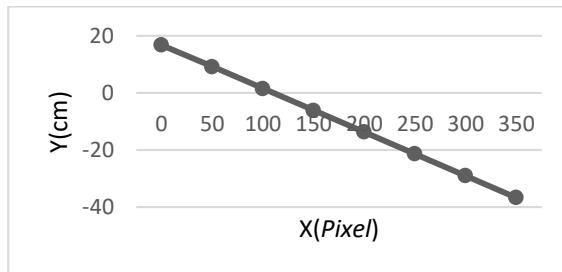
3.5.2 Pengambilan Data Latih *Yaw*

Pengambilan data latih ini bertujuan untuk membuat fungsi regresi linier dalam penentuan gerak berputar drone atau biasa disebut *yaw*. Masukan fungsinya adalah nilai *center X* dari *bounding rectangle* dan keluarannya berupa nilai gerak *yaw*. Fungsi regresi linear berdasarkan 41 data latih *pitch* adalah $y = 0.083784352(x) - 27.70652429$ dan menghasilkan grafik fungsi seperti pada gambar 9.

Gambar 9 Grafik Fungsi Regresi Linier *Yaw*

3.5.3. Pengambilan Data Latih Up/down

Pengambilan data latih ini bertujuan untuk membuat fungsi regresi linier dalam penentuan gerak naik dan turun drone. Masukan fungsinya adalah nilai *center Y* dari *bounding rectangle* dan keluarannya berupa nilai gerak *yaw*. Fungsi regresi linear berdasarkan 41 data latih *pitch* adalah $y = -0.152767413(x) + 16.85594284$ dan menghasilkan grafik fungsi seperti pada gambar 10.

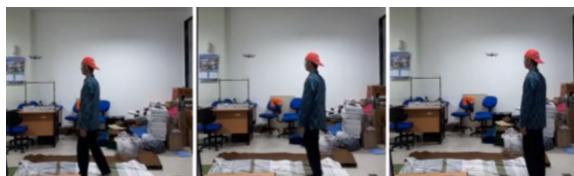


Gambar 10 Grafik Fungsi Regresi Linier Up/down

4. PENGUJIAN

4.1. Pengujian Akurasi Pitch

Nilai *pitch* pada sistem ini didapatkan dengan penerjemahan nilai *height(h)* pada *bounding rectangle* menggunakan regresi linier. Dimana nilai *h* adalah representasi dari ketinggian area objek terdeteksi sehingga bisa menggambarkan jarak objek terdeteksi. Pada pengujian *pitch* ini, *quadcopter* dapat mengikuti pergerakan topi yang dikenakan orang. Orang yang mengenakan topi merah bergerak maju dan mundur lalu *quadcopter* dapat mengikutinya dengan jarak aman yaitu sekitar 1 meter secara *real-time*. Dokumentasi pengujian *pitch* ditunjukkan pada gambar 11.



Gambar 11 Dokumentasi Pengujian Pitch

Tabel 1 Pengujian Akurasi Pitch

No.	X (pixel)	Y (cm)	Aktual (cm)	Error(%)
1	57	123.13	100	23.13
2	55	134.63	100	34.63
3	35	249.57	200	49.57
4	34	255.32	200	55.32
5	29	284.05	300	15.94
6	28	289.80	300	10.19
7	26	301.29	300	1.29
8	19	341.52	400	58.47
9	16	358.88	400	41.22
<i>Mean absolute percentage error</i>				16.03

Kolom X (*pixel*) pada tabel 4.1 berisi nilai dari variabel ‘*h*’ pada *bounding rectangle* yang

merupakan masukan pada fungsi regresi linier. Kolom Y (cm) pada tabel 4.1 berisi nilai hasil dari perhitungan regresi linier untuk dikirimkan sebagai perintah kontrol *pitch* kepada DJI Tello Drone. Kolom aktual (cm) pada tabel 4.1 adalah nilai jarak drone dengan topi yang diukur dengan meteran secara manual. Kolom error(%) pada tabel 4.1 berisi persentasi error berdasarkan selisih perbedaan antara nilai prediksi pada kolom Y (cm) dengan nilai sebenarnya pada kolom *actual* (cm) pada tabel 4.1. Pengujian nilai *pitch* ini mendapatkan hasil berupa nilai rata-rata errornya adalah 16%. Sehingga akurasi sistem dalam menentukan nilai *pitch* untuk maju dan mundur *quadcopter* adalah sebesar 84%.

4.2. Pengujian Akurasi Yaw

Nilai *yaw* pada sistem ini didapatkan dengan penerjemahan nilai horizontal x pada *bounding rectangle* menggunakan regresi linier. Dimana nilai x adalah representasi dari letak objek secara horizontal sehingga bisa menggambarkan posisi objek berada di kanan atau di kiri drone. Pada gambar 12 ditampilkan bagaimana DJI Tello Drone dapat mengikuti objek yang bergerak ke kiri, ke kanan atau memutar drone.



Gambar 12 Dokumentasi Pengujian Yaw

Tabel 2 Pengujian Akurasi Yaw

No.	X (pixel)	Y (°)	Aktual (°)	Error(%)
1	320	-0.89	0	0.89
2	445	9.57	10	4.22
3	185	-12.20	-10	22.06
4	75	-21.42	-20	7.11
5	565	19.63	20	1.84
6	328	-0.22	0	0.22
7	444	9.49	10	5.06
8	566	19.71	20	1.42
9	211	-10.02	-10	0.28
<i>Mean absolute percentage error</i>				6.00

Tabel 2 di atas adalah pengujian akurasi *yaw*. Kolom X (*pixel*) pada tabel 2 berisi nilai dari variabel ‘*center X*’ pada *bounding rectangle* yang merupakan masukan pada fungsi regresi linier. Kolom Y(°) pada tabel 2 berisi nilai hasil dari perhitungan regresi linier untuk dikirimkan sebagai perintah kontrol *yaw* kepada DJI Tello Drone. Kolom aktual (°) pada tabel 2 adalah nilai sudut drone dengan topi yang diukur dengan busur derajat secara manual. Kolom error(%) pada tabel 2 berisi persentasi error berdasarkan selisih perbedaan antara nilai prediksi pada kolom Y

(°) dengan nilai sebenarnya pada kolom *actual* (°) pada tabel 2. Pengujian nilai *yaw* ini mendapatkan hasil berupa nilai rata-rata errornya adalah 6%. Sehingga akurasi sistem dalam menentukan nilai *yaw quadcopter* adalah sebesar 94%.

4.3. Pengujian Akurasi *Up/Down*

Nilai *Up/down* pada sistem ini didapatkan dengan penerjemahan nilai *vertical y* pada *bounding rectangle* menggunakan regresi linier. Dimana nilai *y* adalah representasi dari ketinggian posisi *object* terdeteksi sehingga bisa menggambarkan tinggi atau rendahnya objek terdeteksi. Pada gambar 13 ditampilkan bagaimana DJI Tello Drone dapat mengikuti objek yang bergerak naik dan turun.



Gambar 13 Dokumentasi Pengujian *Up/down*

Tabel 3 Pengujian Akurasi *Up/down*

No.	X (pixel)	Y (cm)	Aktual (cm)	Error(%)
1	97	2.03	0	2.03
2	263	-23.32	-20	16.60
3	72	5.85	5	17.13
4	12	15.02	15	0.15
5	45	9.98	10	0.18
6	288	-27.14	-25	8.56
7	180	-10.64	-10	6.42
8	221	-16.90	-15	12.70
9	320	-32.02	-30	6.76
<i>Mean absolute percentage error</i>				8.56

Tabel 3 di atas adalah pengujian akurasi *up/down*. Kolom X (pixel) pada tabel 3 berisi nilai dari variabel ‘center Y’ pada *bounding rectangle* yang merupakan masukan pada fungsi regresi linier. Kolom Y(cm) pada tabel 3 berisi nilai hasil dari perhitungan regresi linier untuk dikirimkan sebagai perintah kontrol *yaw* kepada DJI Tello Drone. Kolom aktual (cm) pada tabel 3 adalah nilai selisih ketinggian drone dengan topi yang diukur dengan meteran secara manual. Kolom error(%) pada tabel 3 berisi persentasi error berdasarkan selisih perbedaan antara nilai prediksi pada kolom Y (cm) dengan nilai sebenarnya pada kolom *actual* (cm) pada tabel 3. Pengujian nilai *up/down* ini mendapatkan hasil berupa nilai rata-rata errornya adalah 8,5%. Sehingga akurasi sistem dalam menentukan nilai *up/down* untuk naik dan turun *quadcopter* adalah sebesar 91,5%.

4.4. Pengujian Waktu Komputasi

Pengujian waktu komputasi dilakukan dengan membandingkan antara nilai waktu pada saat dimulainya proses iterasi dengan nilai waktu pada saat proses iterasi selesai. Sampel yang digunakan untuk menemukan rata-rata waktu komputasi berjumlah 30 sampel.

Tabel 4 Pengujian Waktu Komputasi

No.	Waktu Komputasi (Detik)
1	0.194
2	0.156
3	0.191
4	0.156
5	0.203
6	0.174
7	0.196
8	0.175
9	0.185
10	0.183
11	0.172
12	0.164
13	0.169
14	0.149
15	0.180
16	0.172
17	0.159
18	0.161
19	0.161
20	0.193
21	0.150
22	0.184
23	0.175
24	0.132
25	0.148
26	0.214
27	0.191
28	0.192
29	0.188
30	0.159
Rata-rata	0.1758

Dari pengujian waktu komputasi di tabel 4, didapatkan hasil berupa nilai rata-rata waktu komputasinya adalah 0.1758 detik atau 5-6 frame per detik. Dengan waktu komputasi yang didapat, dalam implementasinya drone sudah dapat bergerak secara real time dengan baik.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Pada penelitian Sistem *Object tracking* pada *Quadcopter* Menggunakan Segmentasi Citra dengan Deteksi Warna HSV dan Metode Regresi Linier Berbasis *Raspberry pi*, telah menghasilkan kesimpulan bahwa:

1. sistem yang dibuat dapat melakukan pengolahan citra untuk mendeteksi objek dengan segmentasi pada warna HSV pada video langsung. Gambar yang diambil dari kamera drone dikirimkan melalui jaringan nirkabel hingga diproses oleh citra oleh *raspberry pi* untuk dapat mendeteksi objek

- sehingga drone dapat melakukan *object tracking*.
2. Pergerakan drone ditentukan oleh prediksi jarak, sudut, dan ketinggian objek berdasarkan regresi linier yang dihasilkan dari 123 data latih. Hasil deteksi sistem juga cukup sesuai dengan implementasi pergerakan drone. Hal ini diuji dengan 9 data uji di masing-masing perhitungan regresi linier gerak drone yaitu *pitch*, *yaw*, dan *up/down*. Akurasi dari perhitungan gerak *pitch* adalah 84%, akurasi dari perhitungan gerak *yaw* adalah 94%, dan akurasi dari perhitungan gerak *up/down* adalah 91,5%.
 3. Adapun kecepatan waktu komputasi sistem dalam menyelesaikan 1 proses iterasi adalah 0.175829662 detik per frame atau 5-6 frame per detik.

5.2. Saran

Pada penelitian Sistem *Object tracking* pada *Quadcopter* Menggunakan Segmentasi Citra dengan Deteksi Warna HSV dan Metode Regresi Linier Berbasis *Raspberry pi*. Sistem dibuat dengan tujuan hanya mendeteksi objek dan mengikutinya. Saran dari peneliti untuk pengembangan penelitian oleh peneliti selanjutnya yaitu:

1. bagaimana menambahkan suatu Tindakan reaktif pada sistem seperti deteksi aktivitas, deteksi kecelakaan, atau deteksi lokasi dan sebagainya untuk sistem dapat memberi informasi antisipasi.
2. pendektsian objek langsung pada citra RGB dengan tujuan mendapatkan waktu komputasi yang lebih cepat.
3. Pendektsian pada objek menggunakan metode deteksi bentuk untuk menghindari perubahan cahaya dan gangguan warna pada saat pendektsian objek.

DAFTAR PUSTAKA

LE, M.C., LE, M.H., 2019. Human Detection and Tracking for Autonomous Human-following *Quadcopter*, International Conference on System Science and Engineering (ICSSE).

YONG, S.P., YEONG, Y.C., 2018. Human *Object detection* in Forest with Deep Learning based on Drone's Vision, 4th International Conference on Computer and Information Sciences (ICCOINS).

GEORGE, R.P., PRAKASH, V., 2018. *Real-time Human Detection and Tracking Using Quadcopter*. Intelligent Embedded Systems. Lecture Notes in Electrical Engineering, vol 492. Springer, Singapore.

HUTAHAEAN, H.D., WALUYO, B.D., RAIS, M.A., 2019. Teknologi Identifikasi Objek Berbasis *Drone* Menggunakan Algoritma Sift

Citra Digital, Vol. 4 No. 2. Jurnal Teknik Informatika Unika Santo Thomas (JTIUST).

HIKMARIKA, H., IRMAWAN, GUSTINI, A.M., KURNIAWAN, R.D., 2018. Perancangan *Quadcopter* untuk Deteksi Warna Menggunakan Image Processing. Jurnal Rekayasa Mesin (JRM) Universitas Sriwijaya.

PERERA, A.G., AL-NAJI, A., LAW, Y.W., CHAHL, J., 2018. Human Detection and Motion Analysis from a Quadrotor UAV, Volume 405, IOP Conference Series: Materials Science and Engineering.

PANGGABEAN, A.K., SYAHFARIDZAH, A., ARDININGIH, N.A., 2020. Mendeteksi Objek Berdasarkan Warna dengan Segmentasi Warna HSV Menggunakan Aplikasi Matlab, METHOMIKA: Jurnal Manajemen Informatika dan Komputerisasi Akuntansi.

HASTAWAN, A.F., SEPTIANA, R., WINDARTO, Y.E., 2019. Perbaikan Hasil Segmentasi HSV Pada Citra Digital Menggunakan Metode Segmentasi RGB Grayscale, Vol 6 No 1 (2019): Edu Komputika Journal.

Raspberry pi Trading Ltd. 2019. *Raspberry pi 4 Computer Model B*.