

Quadcopter v8: Kaji Pengolahan Citra untuk Misi Terbang Simulasi Pencarian dan Penyelamatan

Ai Siti Aisah^{1,*}, Budi Hartono²

^{1,2}Jurusan Teknik Mesin, Politeknik Negeri Bandung, Bandung 40012
E-mail : ^{1,*}ai.siti.aer20@polban.ac.id; ²buhar@polban.ac.id

ABSTRAK

Kondisi alam dan cuaca yang dinamis menjadi salah satu hambatan yang dialami oleh tim SAR dalam melakukan operasi pendeteksian dan penyelamatan korban di daerah bencana. Oleh sebab itu, penting untuk melengkapi tim SAR dengan suatu perangkat yang dapat membantu pencarian dan penyelamatan di daerah bencana sehingga *survivor* dapat dievakuasi meskipun dalam kondisi yang tidak memungkinkan. Pada penelitian ini dilakukan perakitan *quadcopter*, perancangan program pengolahan citra, dan uji misi terbang simulasi pencarian dan penyelamatan. Simulasi ini dicapai dengan cara merakit *quadcopter* yang dapat terbang dengan stabil, melakukan pemrograman pengolahan citra menggunakan bahasa pemrograman Python dan pustaka OpenCV untuk deteksi objek berupa lingkaran merah, integrasi antara *quadcopter* dan program pengolahan citra, serta uji misi terbang simulasi pencarian dan penyelamatan. Saat uji misi terbang, tangkapan layar yang diambil oleh kamera akan dikirimkan kepada *Ground Control Station* melalui Raspberry Pi sehingga pilot dapat mengambil keputusan berdasarkan apa yang terlihat pada layar. Kamera dengan sudut 0° dapat mendeteksi semua diameter objek yang dipakai. Namun, waktu pendeteksian objek yang paling cepat dicapai dengan menempatkan kamera pada sudut 30°.

Kata Kunci

quadcopter, sistem pengolahan citra, Python, OpenCV, simulasi pencarian dan penyelamatan

1. PENDAHULUAN

Kondisi alam dan cuaca yang dinamis menjadi salah satu hambatan yang dialami oleh tim SAR dalam melakukan operasi pendeteksian dan penyelamatan korban di daerah bencana. Tingkat keberhasilan penyelamatan dalam operasi SAR akan menurun ketika terjadi cuaca yang buruk pada jalur evakuasi sehingga tidak memungkinkan untuk dilakukan operasi penyelamatan. Oleh sebab itu, tim SAR harus memakai suatu perangkat terbang portabel yang dapat membantu pencarian dan penyelamatan di daerah bencana sehingga *survivor* dapat dievakuasi meskipun dalam kondisi yang tidak memungkinkan [1]. Salah satu perangkat yang dapat digunakan adalah *quadcopter* sebagai kendaraan udara tanpa awak yang dapat membantu pencarian dan penyelamatan *survivor* di lokasi yang sulit dijangkau.

Misi terbang *quadcopter* kini dapat dikembangkan dengan menggunakan metode pengolahan citra. Hikmarika et al. menyusun jurnal mengenai perancangan *quadcopter* untuk deteksi warna menggunakan *image processing* atau pengolahan citra. Pada penelitian ini, *quadcopter* akan terbang

pada *waypoint* yang sudah ditentukan untuk mendeteksi objek berupa lingkaran merah dengan memanfaatkan pustaka OpenCV. Jika lingkaran merah terdeteksi, maka Odroid akan mengirimkan perintah kepada Pixhawk sebagai *flight controller* untuk mendarat di atas lingkaran merah tersebut [2].

Hairuissa, L. menyusun tugas akhir mengenai kaji pengolahan citra untuk misi terbang simulasi pencarian dan penyelamatan. Dalam tugas akhir tersebut, *quadcopter* menggunakan pengolahan citra untuk mendeteksi objek berupa lingkaran merah yang kemudian akan menjadi titik *landing* bagi *quadcopter*. Gambar objek tersebut akan diolah oleh Raspberry Pi dengan memanfaatkan pustaka OpenCV. Raspberry Pi kemudian akan memberikan perintah ke Pixhawk sebagai *flight controller* untuk mendarat di atas lingkaran merah tersebut sebelum kemudian *quadcopter* naik dan kembali ke *home* [3].

Pada penelitian ini, *quadcopter* memiliki misi terbang simulasi pencarian dan penyelamatan atau biasa dikenal dengan operasi SAR. *Quadcopter*

yang telah dilengkapi dengan kamera akan mencari objek berupa lingkaran merah pada rute terbang yang telah diatur sebelumnya. Tangkapan layar yang diambil oleh kamera akan dikirimkan kepada *Ground Control Station* melalui Raspberry Pi.

2. LANDASAN TEORI

2.1 Quadcopter

Quadcopter adalah salah satu bentuk UAV (*unmanned aerial vehicle*) atau kendaraan udara tanpa awak yang digerakkan oleh 4 buah motor penggerak yang dikendalikan oleh sebuah *flight controller*. Pada tiap motor dipasang propeler yang berfungsi untuk memberikan gaya angkat pada *quadcopter* sehingga *quadcopter* dapat terbang dan bermanuver. Dua buah motor pada *quadcopter* bergerak searah jarum jam dan dua buah motor lainnya bergerak berlawanan arah jarum untuk saling menghilangkan torsi. Terdapat dua kemungkinan konfigurasi yang digunakan pada *quadcopter*, yaitu konfigurasi X dan konfigurasi + [4].

Berikut merupakan komponen yang digunakan dalam *quadcopter* beserta fungsinya.

Tabel 1. Komponen *quadcopter* beserta fungsinya

Komponen	Fungsi
<i>Frame</i>	Sebagai pembentuk sebuah <i>quadcopter</i> dan tempat untuk menyimpan segala komponen yang ada [5]
<i>Flight Controller</i>	Sebagai perangkat pemrosesan dan pengoperasian sistem UAV <i>quadcopter</i> [6]
<i>Power distribution board</i>	Sebagai pendistribusi daya dan menyediakan jalur yang rapi untuk menghubungkan baterai dengan ESC [7]
<i>Power module</i>	Sebagai pendistribusi sumber tegangan utama yang berasal dari baterai [4]
ESC (<i>Electronic Speed Controller</i>)	Sebagai alat untuk mengonversikan daya dari baterai menjadi sinyal sekuensial yang dapat mengendalikan kecepatan putaran pada motor [4]
Motor	Sebagai pemutar propeler supaya propeler dapat menghasilkan gaya angkat [8]

Propeler	Sebagai penghasil gaya angkat dan pengendali arah terbang <i>quadcopter</i> [5]
GPS <i>module</i>	Sebagai sensor umpan balik <i>quadcopter</i> untuk mengetahui perbandingan antara posisi <i>quadcopter</i> dengan posisi tujuan. [4]
Radio telemetri	Sebagai alat komunikasi antara <i>quadcopter</i> ketika terbang dengan stasiun <i>ground control</i> [4]
<i>Receiver</i>	Sebagai penerima perintah dari <i>remote control</i> [4]
<i>Remote control</i>	Sebagai alat yang digunakan untuk mengoperasikan sebuah <i>quadcopter</i> dari jarak jauh [5]
Baterai	Sebagai sumber daya utama <i>quadcopter</i> [7]
<i>Charger</i>	Sebagai pengisi daya baterai

2.2 Pengolahan Citra

Pengolahan citra adalah sebuah teknik tertentu yang digunakan untuk mengolah sinyal input berupa gambar untuk ditransformasikan menjadi sebuah *output* berupa bentuk gambar yang kualitasnya lebih baik. Pengolahan citra dilakukan dengan cara menganalisis dan memanipulasi gambar untuk memperbaiki kesalahan data input gambar akibat transmisi dan akuisisi sinyal sehingga akan meningkatkan kualitas gambar agar lebih mudah diinterpretasikan oleh sistem penglihatan manusia [9].

Langkah utama yang dilakukan dalam pengolahan citra yaitu [10]:

1. Mengimpor gambar melalui alat akuisisi.
2. Menganalisis dan memanipulasi gambar.
3. Mengeluarkan *output* berupa gambar atau laporan berdasarkan analisis gambar tersebut.

2.3 Misi Terbang Pencarian dan Penyelamatan

Misi terbang dalam penelitian ini yaitu menerapkan simulasi pencarian dan penyelamatan atau biasa dikenal dengan operasi SAR. *Quadcopter* yang telah dilengkapi dengan kamera akan mencari objek berupa lingkaran merah pada rute terbang yang telah diatur sebelumnya. Tangkapan layar yang diambil oleh kamera akan dikirimkan kepada *Ground Control Station* melalui Raspberry Pi. Jika pada *Ground Control Station* terlihat bahwa kamera mendeteksi objek lingkaran merah, pilot akan memberikan perintah

melalui *remote control* untuk *landing* di atas objek tersebut. Setelah *landing*, *quadcopter* akan diam selama beberapa detik sebelum kemudian naik dan kembali ke *home*. Jika pada *Ground Control Station* terlihat bahwa kamera tidak mendeteksi objek, maka *quadcopter* akan terbang menuju titik akhir *waypoint* yang telah ditentukan sebelum kemudian kembali lagi ke *home*.

3. METODOLOGI

3.1 Perakitan *Quadcopter*

Perakitan *quadcopter* dilakukan dengan cara menyusun dan menggabungkan komponen-komponen *quadcopter* menjadi sebuah *quadcopter* yang siap untuk diterbangkan. *Mandatory hardware* yang ada pada *quadcopter* seperti *accelerometer*, *compass*, radio, dan ESC kemudian dikalibrasi melalui *Mission Planner*. Untuk memastikan bahwa komponen *quadcopter* sudah berfungsi dengan baik serta proses kalibrasi sudah dilakukan dengan benar, dilakukan uji terbang manual. Uji terbang PID autotune kemudian dilakukan untuk meningkatkan kemampuan *quadcopter* saat terbang sehingga *quadcopter* dapat terbang dengan lebih andal dan stabil.

3.2 Pemrograman Pengolahan Citra

Pemrograman pengolahan citra bertujuan agar kamera pada *quadcopter* dapat mendeteksi objek berupa lingkaran merah yang akan menjadi titik *landing* bagi *quadcopter*. Pemrograman pengolahan citra dilakukan dengan cara membuat skrip kode menggunakan bahasa pemrograman Python dan pustaka OpenCV. Skrip kode yang digunakan kemudian dimasukkan ke dalam Raspberry Pi agar kamera yang ada pada *quadcopter* dapat melakukan deteksi objek dan tangkapan layar kamera dapat dikirimkan ke *Ground Control Station*.

Setiap warna memiliki jangkauan nilai HSV yang berbeda sehingga harus dilakukan percobaan untuk mencari nilai HSV dari warna yang akan dideteksi. Warna yang digunakan sebagai objek *landing quadcopter* adalah warna merah. Akan tetapi, warna merah memiliki banyak jenis sehingga jangkauan nilai HSV warna merah juga beragam. Untuk mengetahui jangkauan nilai HSV warna merah pada objek *landing quadcopter*, dapat dilakukan dengan metode *trackbar*. Jangkauan nilai HSV warna merah yang sudah

didapatkan kemudian dimasukkan ke skrip kode *real-time detection*.

3.3 Integrasi Perangkat Keras dan Program Pengolahan Citra

Program pengolahan citra yang sudah dibuat ini disimpan pada Raspberry Pi yang dipasang pada *quadcopter*. Raspberry Pi sebagai perangkat pengolahan citra memberikan tampilan video stream yang ditangkap oleh kamera kepada *Ground Control Station* sehingga pilot dapat mengambil keputusan berdasarkan apa yang terlihat di layar.

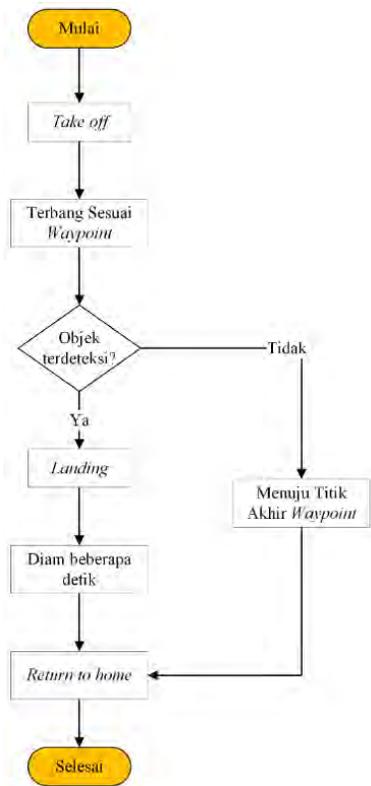
Untuk melakukan integrasi perangkat keras dan program pengolahan citra, dibutuhkan beberapa *software* yaitu:

1. Raspberry Pi Imager untuk instal OS Raspberry Pi.
2. Fing untuk mencari IP Address Raspberry Pi.
3. VNC Viewer untuk mengakses tampilan Raspberry Pi pada laptop.

Untuk melakukan pengolahan citra, Raspberry Pi harus mendapatkan daya yang cukup dengan cara menyambungkan GPIO Raspberry Pi dengan UBEC yang dipasang pada *power distribution board quadcopter*. Setelah memasang Raspberry Pi dengan UBEC, lakukan pemasangan Raspberry Pi dengan komponen pengolahan citra lainnya yaitu kamera melalui modul HDMI to CSI.

3.4 Uji Misi Terbang Pencarian dan Penyelamatan

Uji misi terbang simulasi pencarian dan penyelamatan digambarkan pada diagram alir di bawah ini.



Gambar 1. Diagram alir uji misi terbang

Pada uji misi terbang ini, *quadcopter* yang telah dilengkapi dengan kamera akan mencari objek pada rute terbang yang telah diatur sebelumnya. Tangkapan layar yang diambil oleh kamera akan dikirimkan kepada *Ground Control Station* melalui Raspberry Pi. Jika pada *Ground Control Station* terlihat bahwa kamera mendeteksi objek lingkaran merah, pilot akan memberikan perintah melalui *remote control* untuk *landing* di atas objek tersebut. Setelah *landing*, *quadcopter* akan diam selama beberapa detik sebelum kemudian naik dan kembali ke *home*. Jika pada *Ground Control Station* terlihat bahwa kamera tidak mendeteksi objek, maka *quadcopter* akan terbang menuju titik akhir *waypoint* yang telah ditentukan sebelum kemudian kembali lagi ke *home*.

4. HASIL DAN PENGUJIAN

Pada bagian ini dijelaskan hasil pengujian dari sistem pengolahan citra dan uji misi terbang yang telah dilakukan.

4.1 Hasil Perakitan *Quadcopter*

Komponen-komponen *quadcopter* telah dirakit menjadi sebuah *quadcopter* yang siap untuk diterbangkan. Setelah selesai dirakit, *mandatory*

hardware dari *quadcopter* seperti *accelerometer*, *compass*, *radio*, dan *ESC* kemudian dikalibrasi. Untuk mengecek fungsi komponen *quadcopter* dan keberhasilan kalibrasi, dilakukan uji terbang manual. Uji terbang *PID autotune* kemudian dilakukan untuk meningkatkan kemampuan *quadcopter* saat terbang sehingga *quadcopter* dapat terbang dengan lebih andal dan stabil.



Gambar 2. Hasil perakitan *quadcopter*

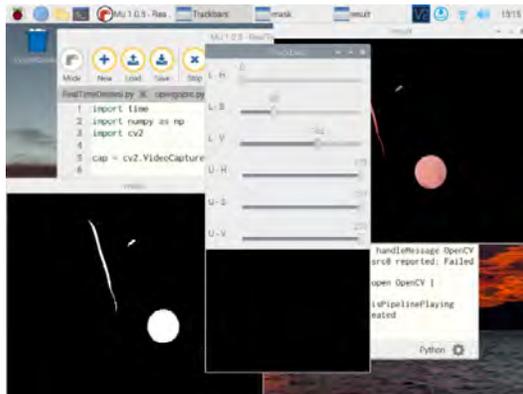
4.2 Hasil Pengolahan Citra

Untuk mendeteksi objek berupa lingkaran berwarna merah, dilakukan *trackbar* nilai HSV warna merah secara *real-time*. *Trackbar* dilakukan dengan cara membuka tampilan Raspberry Pi melalui VNC Viewer kemudian me-*running* kode script *real-time trackbar* pada IDE yang ada. IDE akan menampilkan jendela *trackbar* berupa nilai L-H (*lower hue*), L-S (*lower saturation*), L-V (*lower value*) dan U-H (*upper hue*), U-S (*upper saturation*), dan U-V (*upper value*) beserta 2 jendela kamera yaitu jendela *mask* dan *result*.

Ketika me-*running* skrip kode *trackbar*, maka akan terdapat jendela *trackbar* yang terdiri dari nilai L-H dengan jangkauan 0-179, U-H dengan jangkauan 0-179, L-S dengan jangkauan 0-255, U-S dengan jangkauan 0-255, L-V dengan jangkauan 0-255 dan U-V dengan jangkauan 0-255. Nilai *lower* dan *upper* HSV ini dapat disesuaikan dengan cara menggeser *slider* pada jendela *trackbar* sampai jendela kamera hanya menunjukkan warna yang akan dideteksi.

Nilai HSV pertama digunakan untuk uji misi terbang pertama ketika *quadcopter* terbang untuk mendeteksi lingkaran merah dengan diameter 30 cm dan 60 cm pada ketinggian 5 meter. Pencarian nilai HSV pertama dilakukan ketika siang hari dengan cara memegang *quadcopter* di atas lingkaran merah dan program pengolahan citra dilakukan pada Raspberry Pi yang mendapatkan daya dari power bank. Hal ini dilakukan untuk

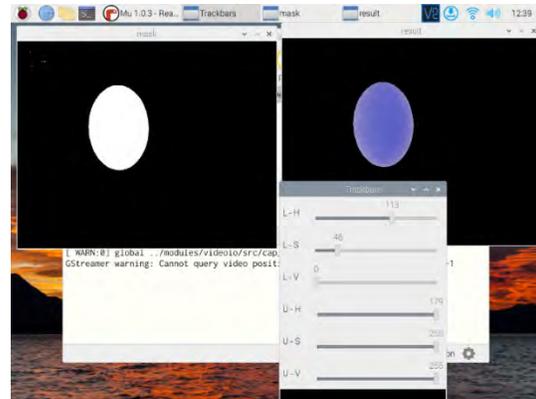
menghemat daya baterai sehingga baterai dapat dipakai untuk beberapa kali uji misi terbang.



Gambar 3. *Trackbar* nilai HSV pertama

Gambar 3 merupakan tampilan *real-time trackbar* untuk mencari nilai HSV pada uji misi terbang pertama. Nilai HSV untuk uji misi terbang pertama ini memiliki nilai L-H sebesar 0, L-S sebesar 69, L-V sebesar 162, U-H sebesar 179, U-S sebesar 255, dan U-V sebesar 255. Setelah mendapatkan nilai HSV dari warna merah, pendeteksian objek pada saat uji misi terbang dilakukan dengan cara memasukkan nilai HSV ke dalam kode script *real-time detection* pada baris `np.array` sehingga menjadi `'lower_hsv = np.array([0, 69, 162])'` dan `'upper_hsv = np.array([179,255,255])'`.

Nilai HSV kedua digunakan untuk uji misi terbang kedua ketika *quadcopter* terbang untuk mendeteksi lingkaran merah dengan diameter 40 cm dan 50 cm pada ketinggian 5 meter. Pencarian nilai HSV kedua dilakukan ketika siang hari pada hari yang berbeda. Intensitas cahaya saat dilakukan uji misi terbang pertama dan kedua berbeda, sehingga nilai HSV kembali dicari ketika akan dilakukan uji misi terbang kedua. Pencarian nilai HSV dilakukan dengan cara memegang *quadcopter* di atas lingkaran merah dan program pengolahan citra dilakukan pada Raspberry Pi yang mendapatkan daya dari power bank. Hal ini dilakukan untuk menghemat daya baterai sehingga baterai dapat dipakai untuk beberapa kali uji misi terbang.



Gambar 4. *Trackbar* nilai HSV kedua

Gambar 4 merupakan tampilan *real-time trackbar* untuk mencari nilai HSV pada uji misi terbang kedua. Nilai HSV untuk uji misi kedua ini memiliki nilai L-H sebesar 113, L-S sebesar 46, L-V sebesar 0, U-H sebesar 179, U-S sebesar 255, dan U-V sebesar 255. Setelah mendapatkan nilai HSV dari warna merah, pendeteksian objek pada saat uji misi terbang dilakukan dengan cara memasukkan nilai HSV ke dalam kode script *real-time detection* pada baris `np.array` sehingga menjadi `'lower_hsv = np.array([113, 46, 0])'` dan `'upper_hsv = np.array([179,255,255])'`

Pada uji misi terbang kedua, layar yang menampilkan jendela kamera ketika melakukan *real-time trackbar* memperlihatkan bahwa lingkaran merah memiliki warna biru. Hal ini dikarenakan peletakan perangkat keras pengolahan citra yaitu modul HDMI to CSI tidak tepat sehingga warna pada tampilan layar berbeda dengan apa yang ditangkap oleh kamera. Namun, nilai HSV tersebut tetap mempresentasikan nilai HSV warna merah karena saat dilakukan uji misi terbang dengan nilai HSV tersebut, lingkaran merah terdeteksi.

4.3 Hasil Uji Misi Terbang

Uji misi terbang pertama dilakukan pada siang hari di lapangan terbuka berumput hijau. *Waypoint* pada uji misi terbang pertama adalah 25 meter. Pada *waypoint* ini ditempatkan sebuah lingkaran merah, tepatnya 15 meter dari titik *take off*. Uji misi terbang pertama dilakukan untuk mendeteksi lingkaran merah dengan diameter 30 cm dan 60 cm pada ketinggian 5 meter.

Uji misi terbang kedua dilakukan pada siang hari di lapangan terbuka berumput hijau. *Waypoint* pada uji misi terbang kedua adalah 25 meter. Pada

waypoint ini ditempatkan sebuah lingkaran merah, tepatnya 15 meter dari titik *take off*. Uji misi terbang kedua dilakukan untuk mendeteksi lingkaran merah dengan diameter 40 cm dan 50 cm pada ketinggian 5 meter.

Pendeteksian objek dilakukan menggunakan kamera Go Pro Hero 4 dengan sudut pandang *wide angle* yang memberikan cakupan sudut pandang sebesar 170°. Berikut ini merupakan tabel hasil uji misi terbang pertama dan kedua.

Tabel 2. Hasil uji misi terbang

Diameter (cm)	Sudut Kamera (°)	Hasil Deteksi	Waktu Deteksi (s)
30	0	Terdeteksi	10
	30	Tidak	-
	60	Tidak	-
40	0	Terdeteksi	9
	30	Terdeteksi	7
	60	Tidak	-
50	0	Terdeteksi	7
	30	Terdeteksi	5
	60	Terdeteksi	6
60	0	Terdeteksi	30
	30	Terdeteksi	18
	60	Terdeteksi	20

Dari tabel 2, terlihat bahwa sudut 0° dapat mendeteksi lingkaran dengan diameter 30 cm, 40 cm, 50 cm, dan 60 cm. Sudut 30° tidak dapat mendeteksi objek dengan diameter yang kecil yaitu 30 cm. Hal ini dikarenakan sudut 30° memiliki bidang pandang yang lebih luas sehingga objek dalam *frame* terlihat kecil dan dapat menyebabkan algoritma sulit untuk mengenali objek. Begitu pula dengan sudut 60°, sudut ini tidak dapat mendeteksi objek dengan diameter 30 cm dan 40 cm karena memiliki bidang pandang yang lebih luas dari sudut 30° sehingga objek dalam *frame* terlihat lebih kecil dan dapat menyebabkan algoritma sulit untuk mengenali objek.

Sudut 30° dapat mendeteksi objek dengan waktu yang lebih singkat dibandingkan dengan sudut 0° dikarenakan sudut 30° memiliki bidang pandang yang lebih luas sehingga dengan jarak yang sama, objek akan muncul dalam *frame* dalam waktu yang lebih cepat. Sementara itu, sudut 30° mendeteksi objek dengan waktu yang lebih singkat dari 60° dikarenakan sudut 60° memiliki cakupan lebih luas sehingga proses pengolahan citra membutuhkan waktu yang lebih lama karena algoritma harus memeriksa lebih banyak area.

5. KESIMPULAN

Quadcopter dapat melakukan pengolahan citra untuk mendeteksi objek berupa lingkaran merah. Pengolahan citra ini dilakukan dengan bantuan bahasa pemrograman Python dan pustaka OpenCV. Kode *script* yang digunakan untuk mendeteksi objek dimasukkan ke dalam Raspberry Pi yang dipasang pada *quadcopter*. Raspberry Pi memungkinkan akses pada kamera Go Pro Hero 4 sebagai web cam sehingga pengolahan citra dapat dilakukan secara *real-time*.

Untuk melakukan simulasi pencarian dan penyelamatan, *quadcopter* yang telah dilengkapi dengan kamera akan mencari objek pada rute terbang yang telah diatur sebelumnya. Tangkapan layar yang diambil oleh kamera akan dikirimkan kepada *Ground Control Station* melalui Raspberry Pi. Jika pada *Ground Control Station* terlihat bahwa kamera menangkap objek berupa lingkaran merah, maka pilot akan memberikan input pada *remote control* agar *quadcopter landing* di atas lingkaran merah tersebut. Jika tidak, *quadcopter* akan terbang menuju titik akhir *waypoint* dan kembali ke *home*.

Sudut 0° dapat mendeteksi lingkaran dengan diameter 30 cm, 40 cm, 50 cm, dan 60 cm. Namun, Sudut 30° dapat mendeteksi objek dengan waktu yang lebih singkat dibandingkan dengan sudut 0° dikarenakan sudut 30° memiliki bidang pandang yang lebih luas sehingga dengan jarak yang sama, objek akan muncul dalam *frame* dalam waktu yang lebih cepat. Sementara itu, sudut 30° mendeteksi objek dengan waktu yang lebih singkat dari 60° dikarenakan sudut 60° memiliki cakupan lebih luas sehingga proses pengolahan citra membutuhkan waktu yang lebih lama karena algoritma harus memeriksa lebih banyak area.

DAFTAR PUSTAKA

- [1] K. B. Yogha and R. Lipikorn, Sistem Penginderaan Berbasis UAV Untuk Membantu Operasi Pencarian dan Penyelamatan Korban Kecelakaan di Wilayah Pegunungan, vol. 06, 2017, pp. 293-308.
- [2] H. Hikmarika, I. G. A. Morgana and R. Kurniawan, Perancangan *Quadcopter* untuk Deteksi Warna Menggunakan Image Processing, vol. 18, 2018, pp. 1-8.
- [3] L. Hairunnisa, "*Quadcopter* dengan Sistem Pengolahan Citra: Kaji Pengolahan Citra untuk Misi Terbang Simulasi Pencarian dan Penyelamatan," Politeknik Negeri Bandung, Bandung, 2020.

- [4] A. K. Rahman, H. Supriyanto and T. Meizinta, Rancang Bangun dan Implementasi Sistem Kendali *Quadcopter* melalui Jaringan Internet Berbasis Lokasi Menggunakan Smartphone Android, Bandung, 2018.
- [5] A. F. Harista and S. Nuryadi, Sistem Navigasi *Quadcopter* dan Pemantauan Udara, vol. 01, 2018, pp. 1-22.
- [6] S. Sabikan and S. W. Nawawi, "Open-Source Project (OSPs) Platform for Outdoor *Quadcopter*," Journal of Advanced Research Design, vol. 24, no. 1, pp. 13-27, 2016.
- [7] Dwi, "Mengenal Prinsip, Jenis, Bagian dan Komponen Drone," Idmetafora, 30 Agustus 2022. [Online]. Available: <https://idmetafora.com/news/read/997/Mengenal-Prinsip-Jenis-Bagian-dan-Komponen-Drone.html>. [Accessed 7 Januari 2023].
- [8] Heriansyah, "How to Make a *Quadcopter* Drone and Components List," Autodesk Instructables, [Online]. Available: <https://www.instructables.com/How-to-Make-a-Quadcopter-Drone-and-Components-List/>. [Accessed 7 Januari 2023].
- [9] M. Latif and H. Budiarto, "PERANCANGAN SISTEM AUTONOMOUS *QUADCOPTER*," Seminar Nasional Sains dan Teknologi, pp. 1-5, 2014.
- [10] N. Kumar, "Digital Image Processing Basics," 30 November 2021. [Online]. Available: <https://www.geeksforgeeks.org/digital-image-processing-basics/>. [Accessed 7 Januari 2022].