

Navigasi *Quadcopter* Berbasis *ArUco Marker* dengan OpenCV

Taufiqul Hakim¹, Ardy Seto Priambodo¹

¹Departemen Teknik Elektro dan Elektronika, Universitas Negeri Yogyakarta; taufiqulhakim628@gmail.com

*Korespondensi: ardyseto@uny.ac.id

Abstract – *It is certain that companies engaged in the production of goods always monitor the stock of goods in the warehouse. Accuracy in calculating stock is needed to obtain accurate actual data. The use of human power to handle this is certainly not optimal if the stock of goods is at a certain height. With these conditions, a technology is needed that is able to overcome this, especially how the monitoring equipment can move from start to finish with certain height variations. Driven by technological advances, UAVs (Unmanned Aerial Vehicles) are now equipped with supporting devices that enable them to be completely controlled autonomously. In general, UAV (Unmanned Aerial Vehicle) can be interpreted as an aerial vehicle that operates in the air like a quadcopter. Quadcopters are unmanned aircraft with four motors that allow them to take off and land vertically or what is usually called VTOL (Vertical Take-Off Landing). Utilizing a quadcopter can be a solution design that combines integrated automatic control. The method used in this research is the Engineering Design Process which includes exploring problems, considering solutions, design, development, testing and evaluation. In this research, four experiments were carried out with different trajectories and different altitudes according to the problems raised. There were four experiments and different altitude requests aimed at seeing the angle error and altitude error for each ArUco marker detection. The research results show that there are different angular errors and the smallest average angular error is 2.6°. There is also a difference in the altitude results in the 4th experiment from the demand figure, namely 3 meters. These results are influenced by wind disturbances that push the quadcopter body, the type of camera used, and telemetry as a data transmission medium.*

Keywords – *ArUco marker, tracking inventory, quadcopter*

Intisari – Perusahaan yang bergerak di bidang produksi barang sudah dipastikan selalu memantau stok barang di gudang. Keakuratan penghitungan stok barang sangat dibutuhkan untuk mendapatkan data aktual yang tepat. Pemanfaatan tenaga manusia untuk menangani hal tersebut tentu kurang maksimal jika keberadaan stok barang berada di ketinggian tertentu. Dengan kondisi tersebut, dibutuhkan suatu teknologi yang mampu mengatasi hal tersebut, terkhusus bagaimana alat pemantau dapat bergerak dari awal hingga akhir dengan variasi ketinggian tertentu. Didorong dengan kemajuan teknologi, UAV (*unmanned aerial vehicle*) kini sudah dilengkapi perangkat pendukung yang memungkinkan dapat sepenuhnya dikendalikan secara otonomi. Secara umum, UAV dapat diartikan sebagai sebuah wahana udara yang beroperasi di udara seperti halnya *quadcopter*. *Quadcopter* adalah pesawat tanpa awak dengan empat motor yang memungkinkan mereka untuk lepas landas dan mendarat secara vertikal atau biasa disebut dengan VTOL (*vertical take-off landing*). Pemanfaatan *quadcopter* dapat menjadi salah satu rancangan solusi dengan menggabungkan kendali otomatis yang terpadu. Metode yang digunakan dalam penelitian ini adalah *Engineering Design Process* yang meliputi penggalian permasalahan, pertimbangan solusi, desain, pengembangan, uji coba, dan evaluasi. Pada penelitian ini dilakukan empat percobaan dengan lintasan yang berbeda dan ketinggian (*altitude*) yang berbeda sesuai dengan permasalahan yang diangkat. Adanya empat percobaan dan permintaan *altitude* yang berbeda bertujuan untuk melihat *error* sudut maupun *error altitude* pada setiap deteksi *ArUco marker*. Hasil penelitian menunjukkan bahwa terdapat *error* sudut yang berbeda dan didapatkan rerata *error* sudut paling kecil yakni di angka 2,6°. Terdapat juga perbedaan hasil *altitude* pada percobaan ke-4 dari angka permintaan yakni 3 meter. Hasil ini dipengaruhi oleh gangguan angin yang mendorong badan *quadcopter*, jenis kamera yang digunakan, dan telemetri sebagai media transmisi data.

Kata kunci – *ArUco Marker, tracking inventory, quadcopter*

I. PENDAHULUAN

Proses pemantauan kini sudah merambah ke berbagai bidang, dari yang sebelumnya tidak membutuhkan sistem otomatis hingga saatnya sekarang sudah membutuhkan sebuah sistem yang terpadu. Sehingga data pantauan dapat diakses secara *real-time* seperti perusahaan yang melakukan proses produksi. Perusahaan yang melakukan proses produksi sangat membutuhkan *tracking inventory* [1]. Perusahaan akan mengalami kesulitan saat menjalankan operasionalnya karena ketidakprofesionalan perusahaan dalam mengelola barang.

Perusahaan harus secara efektif melakukan peningkatan keamanan aset dan pengawasan. Semuanya, baik material maupun non-material, akan dihemat dengan sistem kerja efisien. Seperti yang terdapat di gudang milik IKEA yang memiliki tinggi di atas jangkauan manusia pada umumnya, hal ini akan menyulitkan proses *tracking inventory* [2]. Dengan hal ini diperlukan sebuah teknologi yang mampu menjangkau tempat-tempat yang sulit dijangkau manusia. Dalam hal akses ketinggian, *quadcopter* dapat menjadi salah satu solusi untuk membantu meningkatkan efisiensi monitoring aset perusahaan.

Quadcopter adalah pesawat tanpa awak dengan empat motor yang memungkinkan mereka untuk lepas landas dan mendarat secara vertikal atau biasa disebut dengan VTOL (*vertical take-off landing*) [3]. Setiap motor memiliki kecepatan yang berbeda, yang memungkinkan mereka untuk bermanuver. *Quadcopter* temuan pertama ini diberi nama Omnichen 2 setelah ditemukan oleh Etienne Mahakuasa pada tahun 1920. Penemuan pertama ini berhasil melakukan 1000 penerbangan dengan jarak maksimal 360 meter dari titik *home*. Dr. George E. Bothezat menciptakan Omnichen 2, yang mengubah *frame quadcopter* dari yang sebelumnya memiliki sayap menjadi yang sekarang [4].

Secara umum, UAV (*unmanned aerial vehicle*) dapat diartikan sebagai sebuah wahana udara yang beroperasi di udara. Didorong dengan kemajuan teknologi, UAV kini sudah dilengkapi perangkat pendukung yang memungkinkan dapat sepenuhnya dikendalikan secara *autonomous* [5]. Dewasa ini, peningkatan sistem *autonomous* UAV digunakan pada sistem wahana dengan tipe *rotary-wing*, salah satunya yakni *quadcopter* [6]. Saat ini sudah terdapat sistem otomatis dari *quadcopter* ditandai dengan banyaknya modul pemrograman sehingga memungkinkan menggerakkan *quadcopter* berdasarkan parameter yang sudah diatur [7].

ArUco marker dapat dimanfaatkan untuk mengendalikan robot, seperti *quadcopter* karena memiliki ID khusus, sehingga dapat dimanfaatkan sebagai pemandu gerakan suatu robot. ArUco marker adalah repositori atau tempat penyimpanan simbol berbentuk marker yang umum digunakan untuk aplikasi AR (*augmented reality*) berbasis OpenCV [8]. Dikarenakan sudah banyak digunakan dan terdapat banyak referensi, pada penelitian ini digunakan ArUco Marker sebagai objek deteksi agar mempermudah jalannya proses penelitian.

OpenCV merupakan modul pemrograman *open source* yang dapat memproses gambar hingga *computer vision*. OpenCV memiliki modul pemrograman yang dikhususkan untuk mengontrol *quadcopter* yakni DroneKit. DroneKit mendukung berbagai gerakan terbang *quadcopter* seperti *hovering*, *roll*, *pitch*, dan *yaw*. Dengan alasan tersebut, library OpenCV digunakan dalam penelitian.

Pada *quadcopter* terdapat beberapa macam gerakan yaitu *hovering*, *roll*, *pitch*, dan *yaw* [9]. Semua pergerakan *quadcopter* dipengaruhi oleh torsi motor, kecepatan putaran motor, dan momen pada *frame quadcopter* [5]. Gerakan inilah yang akan diperintahkan dari ArUco marker sehingga dapat berjalan sesuai dengan misi terbang yang seluruhnya sudah disediakan oleh OpenCV.

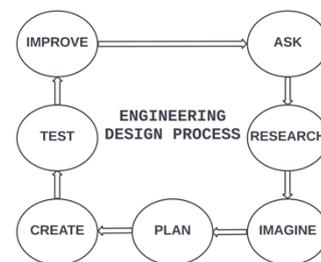
Pada penelitian ini, *quadcopter* akan membaca ArUco marker yang diletakan pada lantai sehingga dapat berjalan secara *autonomous* dengan dipandu gerakannya lewat ArUco marker dan dapat dimonitoring jarak jauh dengan memanfaatkan video *streamer*. Pada penelitian ini akan dilihat parameter-parameter yang akan dievaluasi dari penggunaan *quadcopter* jenis *outdoor* dan penggunaan OpenCV baik dari performa terbang atau performa program

sehingga didapatkan navigasi terbang *quadcopter* yang maksimal.

II. METODOLOGI

Metode *Engineering Design Process* seperti yang ditunjukkan pada Gambar 1 digunakan dikarenakan lebih rinci pada setiap prosesnya. Tahapan pada *Engineering Design Process* sangat melekat pada penelitian-penelitian di bidang teknik yang akan membangun sebuah prototipe. Dengan hal itu, metode *Engineering Design Process* digunakan sebagai landasan perencanaan penelitian.

Tahap perencanaan dilakukan dengan menganalisis kebutuhan *hardware* dan *software*. Kebutuhan *hardware* meliputi perencanaan elektronik dan mekanik. Sedangkan kebutuhan *software* meliputi perencanaan *software* yang akan digunakan dalam pembangunan prototipe baik berupa aplikasi, *firmware*, *library*, hingga algoritma pemrograman.

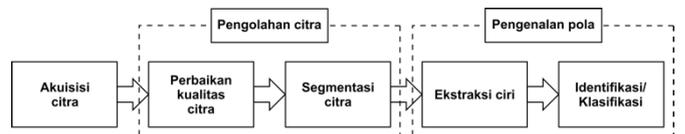


Gambar 1. Metode Penelitian *Engineering Design Process*

A. OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah *library computer vision*. OpenCV dibangun untuk menyediakan infrastruktur umum pada aplikasi *computer vision* [10]. *Computer vision* bekerja dengan menggunakan kemampuan penglihatan manusia untuk melihat dan menganalisis visual [11]. Dengan *library* OpenCV memungkinkan untuk mengolah gambar.

Pengolahan citra (*image processing*) merupakan bidang yang erat kaitannya dengan proses transformasi citra/gambar [12]. Diagram pada Gambar 2 merupakan langkah-langkah yang umumnya dilakukan dalam merancang sebuah sistem *computer vision* [13].



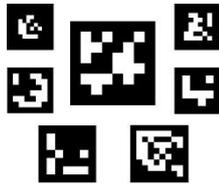
Gambar 2. Tahapan *computer vision*

Diagram yang ditunjukkan pada Gambar 2 merupakan tahapan *image processing* yang berisi 2 tahapan yakni pengolahan citra dan pengenalan pola. Pada tahap awal citra diperbaiki untuk mendapatkan hasil yang baik sehingga dapat dilakukan pengenalan pola. *Computer vision* digunakan dalam penelitian ini dikarenakan terdapat tahapan pengenalan

pola sehingga mampu mengenali objek deteksi dengan pola yang berbeda.

B. ArUco Marker

ArUco marker adalah marker kotak sintetik yang disusun oleh garis tepi hitam lebar dan matriks biner bagian dalam yang membentuk suatu bentuk kode sehingga dapat menentukan pengidentifikasi (ID) [14].



Gambar 3. ArUco marker

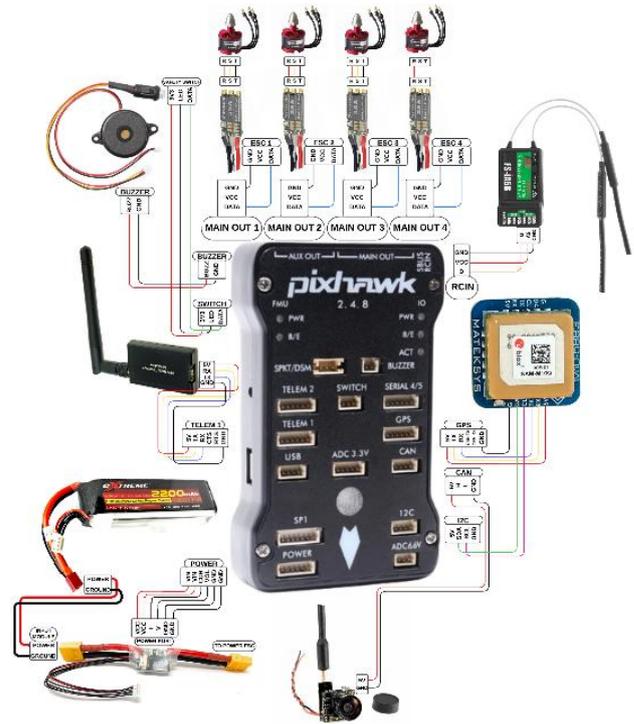
ArUco marker memiliki banyak keunggulan, seperti mudah dibuat, mendeteksi dengan cepat, dan berfungsi dalam berbagai kondisi pencahayaan [15]. Seperti yang terlihat pada Gambar 3, ArUco marker memiliki banyak bentuk piksel bergantung pada ID yang dimiliki. ArUco marker sering digunakan dalam proyek AR, robotika, dan aplikasi berbasis visi. Untuk menggunakan ArUco marker, biasanya diperlukan dua komponen: perangkat lunak pengolahan gambar yang dapat menginterpretasikan informasi dari marker dan kamera yang dapat mendeteksi dan mengenali marker melalui program yang sudah dituliskan.

C. Perencanaan Hardware

1) *Rancangan Elektronik*: Gambar 4 menunjukkan rangkaian komponen elektronik yang menggunakan FC (flight controller) jenis PIXHAWK 2.4.8. FC ini digunakan sebagai pengontrol quadcopter dan akan mengeluarkan perintah untuk menyesuaikan kecepatan motor dengan menggunakan PID yang sudah dikonfigurasi [16]. Untuk kebutuhan navigasi, digunakan GPS dengan tipe MQ10 dari MATEKSYS. Data yang didapatkan dari GPS adalah di antaranya *latitude*, *longitude*, dan *altitude* sebagai acuan gerakan *quadcopter* sehingga dapat menyelesaikan suatu misi [17].

Quadcopter pada penelitian ini menggunakan 4 motor dengan konfigurasi bentuk *frame* yakni Quad-X. Konfigurasi frame X dinilai lebih stabil daripada jenis *frame* lainnya [18]. Karena terdapat 4 motor, maka digunakan juga empat buah ESC (*electronic speed controller*) sebagai pengontrol kecepatan motor. ESC bekerja dengan mengirim sinyal ke motor dengan menerima sinyal PWM dari FC [19].

Digunakan telemetry sebagai pengirim dan penerima data baik dari *ground control* maupun wahana. Prototipe ditenagai oleh baterai Li-Po 2200 mAh 4s untuk menyuplai daya pada semua komponen elektronik. Setiap sel baterai Li-Po memiliki tegangan 3.75V [18]. Sebagai indikator status yang dimiliki FC, digunakan *buzzer* untuk memberikan informasi status dalam bentuk bunyi. Daftar komponen elektronik seperti pada Tabel 1.



Gambar 4. Rangkaian komponen elektronik

Tabel 1. Daftar komponen elektronik

No.	Nama Komponen
1	Pixhawk 2.4.8
2	ESC CYCLONE 35A
3	Motor READYTOSKY 2212 920KV
4	GPS M10Q
5	Baterai 4s 2200mAh
6	FS-I6 dan FS-IA6B RC
7	Micro camera
8	Telemetry 433MHz

2) *Rancangan Mekanik*: Rancangan mekanik terdiri dari 2 perencanaan yakni desain *quadcopter* dan objek deteksi dalam hal ini adalah ArUco marker.

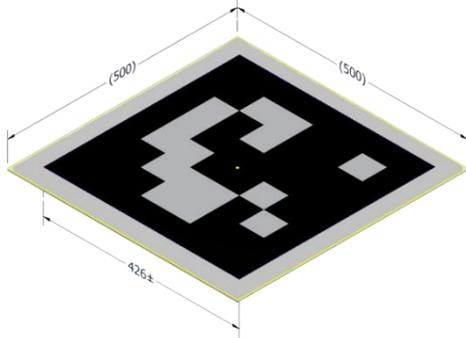


Gambar 5. Desain *quadcopter* F450

Desain pada Gambar 5 merupakan gambaran *quadcopter* dengan konfigurasi Quad-X yang memiliki 4 *propeller* dengan 2 arah putar CW (*clockwise*) dan CCW (*counterclockwise*) [20]. Terdapat perbedaan warna di kedua sisi yakni sisi merah dan putih. Peletakan warna yang berbeda

ini bertujuan untuk mempermudah membaca *heading*/arah *quadcopter*.

Dibutuhkan juga desain objek deteksi sebagai acuan dalam pembuatan prototipe. Objek deteksi menggunakan ArUco *marker* dengan ID yang berbeda. *Marker* membutuhkan alas sehingga membuatnya mampu menopang *quadcopter* ketika *take-off* dan *landing*.



Gambar 6. Desain objek deteksi (ArUco *marker*)

Desain objek deteksi seperti yang ditunjukkan pada Gambar 6 menggunakan alas berjenis *impraboard* dengan ukuran 50 cm x 50 cm. Sedangkan untuk *marker* berukuran 40 cm agar dapat terbaca di ketinggian.

D. Perencanaan Software

Perancangan *software* terbagi menjadi 2 bagian yakni *firmware* dan *control system*. Gambar 7 merupakan pembagian perencanaan *software*.



Gambar 7. Daftar kebutuhan *software*

Pembagian yang ditunjukkan pada Gambar 7 ini bertujuan untuk memudahkan proses analisis kebutuhan *software*. *Quadcopter* pada penelitian ini menggunakan *firmware* ArduPilot. ArduPilot adalah perangkat *open source* yang memungkinkan pengguna mengonfigurasi sistem agar berjalan secara mandiri [17]. Ardupilot dipilih dikarenakan dapat didukung oleh *library* DroneKit yang berfungsi untuk memberikan perintah gerakan *quadcopter* seperti *roll*, *pitch*, *yaw*, *throttle*, dll. [21]. Sedangkan untuk kebutuhan pemrosesan *image detection*, dalam penelitian ini digunakan OpenCV dikarenakan memiliki modul ArUco sendiri dan menggunakan konsep *computer vision* dalam pemrosesan video.

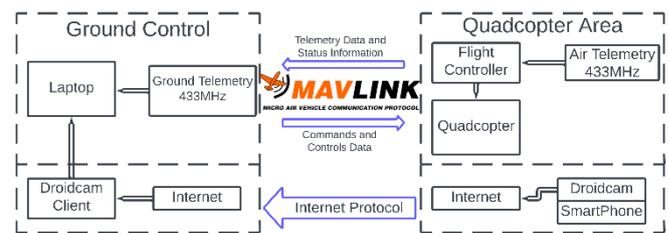
Penulisan program dan konfigurasi *library* digunakan PyCharm sebagai *text editor* pemrograman. PyCharm dipilih dikarenakan memiliki alur konfigurasi yang mudah seperti pengaturan Python *interpreter*, *library*, modul, dll [22].

Dikarenakan sistem *autonomous* pada penelitian ini dikontrol di *ground system*, maka dibutuhkan pengiriman data video dari wahana ke *ground system* agar dapat diolah. Pada penelitian ini digunakan video *streamer* DroidCam. DroidCam digunakan karena memiliki konfigurasi yang mudah dan hanya menggunakan internet sebagai protokol pengiriman data video.

Kedua bagian tersebut dihubungkan oleh suatu protokol komunikasi yakni MAVLink. MAVLink (*Micro Air Vehicle Link*) merupakan salah satu protokol komunikasi yang sering digunakan dalam pembangunan sistem tanpa awak tak terkecuali pengembangan *quadcopter* [23]. MAVLink bekerja dengan mengubah struktur informasi ke dalam bentuk biner dan kemudian diteruskan dalam bentuk baris *transport* yang ada pada jaringan [19].

E. Integrasi Sistem

Integrasi sistem merupakan rancangan penggabungan semua komponen yang akan saling berkomunikasi dengan bertukar atau mengirimkan data.



Gambar 8. Integrasi sistem

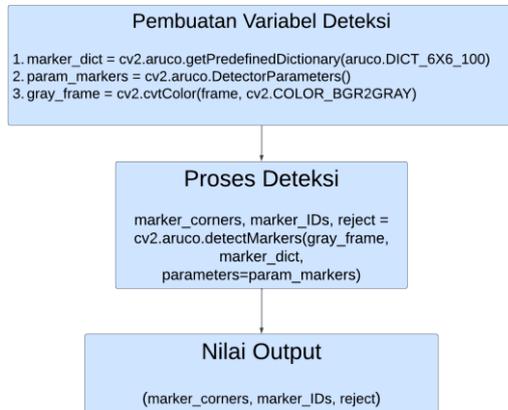
Terdapat 2 bagian desain integrasi sistem berdasarkan protokol komunikasinya. Seperti yang ditunjukkan pada Gambar 8, pada sisi atas menghubungkan antara *ground control* dan wahana/*quadcopter area* dengan menggunakan protokol MAVLink untuk kebutuhan pengiriman perintah gerakan dari *ground control* ke wahana ataupun permintaan data dari wahana. Sedangkan pada sisi bawah menghubungkan *ground control* dengan wahana untuk kebutuhan pengiriman video dari wahana ke *ground control*.

F. Konsep Deteksi ArUco Marker

1) *Deklarasi Program*: Deteksi ArUco *marker* dapat dilakukan dengan menggunakan bahasa pemrograman Python dengan *library* Opencv. Gambar 9 merupakan tahapan konsep deteksi ArUco *marker* dari sisi penulisan program.

Proses deteksi ArUco *marker* memerlukan tiga parameter/variabel seperti yang ditunjukkan pada Gambar 9. Parameter dan contoh *output* dapat dilihat pada Tabel 2. Parameter “*marker_dict*” digunakan untuk menyatakan *library* ArUco *marker* yang digunakan. Parameter “*param_markers*” digunakan untuk mendeteksi parameter yang dibutuhkan seperti data koordinat. Sedangkan

“gray_frame” digunakan untuk membuat tangkapan video menjadi berwarna dengan skala *gray* sesuai tingkat kepekaan warna *marker* saat ditangkap. Parameter-parameter ini kemudian dijadikan bahan untuk proses deteksi dimulai.



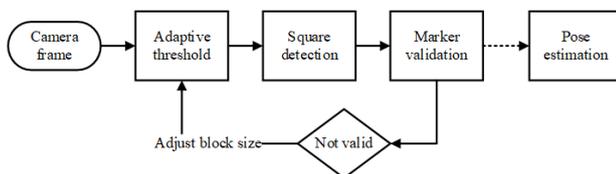
Gambar 9. Alur deteksi objek dalam deklarasi program

Tabel 2. Parameter dan contoh *output*

Parameter	Contoh Output
Marker_dict	Library ArUco (6x6, 4x4, dll)
Param_markers	Koordinat x dan y (0.0 ; 1.0)
Gray_frame	Oalah gambar yang sudah diubah menjadi “gray”

Ketika pada tahap proses deteksi sudah berjalan, maka akan menghasilkan tiga nilai yang disimpan pada masing-masing variabel yakni “marker_corners” (nilai *corners* berisi data koordinat x dan y dari ArUco *marker* yang telah terdeteksi), “marker_ids” (data ids yakni id yang dimiliki objek ArUco *marker* yang dideteksi), dan “reject” (Nilai *rejected* adalah sekumpulan data yang akhirnya ditolak karena datanya tidak dapat diuraikan).

2) *Tahap Image Processing*: Untuk objek memungkinkan terdeteksi dan menghasilkan beberapa nilai diperlukan pemrosesan gambar seperti yang ditunjukkan pada Gambar 10 yakni terdapat tahap *thresholding*, *contour filtering*, *bits extraction*, dan *marker identification*.

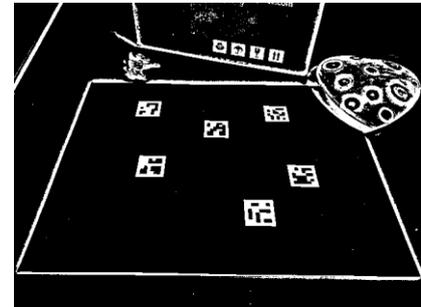


Gambar 10. Tahapan *image processing*

Untuk objek memungkinkan terdeteksi dan menghasilkan beberapa nilai diperlukan pemrosesan gambar seperti yang ditunjukkan pada Gambar 10 yakni terdapat tahap *thresholding*, *contour filtering*, *bits extraction*, dan *marker identification*.

• *Thresholding*

Thresholding merupakan tahapan segmentasi citra yang berfungsi untuk memisahkan antara objek dengan *background* dengan tingkat kecerahan warna berbeda [24], seperti yang ditunjukkan pada Gambar 11. Untuk memungkinkan objek dalam hal ini ArUco *marker* dapat terbaca dan memberikan *return value*.



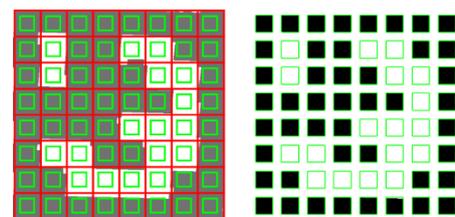
Gambar 11. *Thresholding*

• *Contour Filtering*

Kontur adalah kurva yang menghubungkan suatu titik yang memiliki entitas sama. Setelah *thresholding*, didapatkan hasil sekumpulan titik putih dan hitam yang membentuk pola. Pada tiap titik inilah yang akan digunakan dalam pemrosesan dan akan membuang suatu kontur yang tidak terpakai dalam hal ini suatu titik yang tidak membentuk pola.

• *Bits Extraction*

Bit extraction adalah mengekstraksi kontur yang berwarna putih apakah termasuk ke dalam penanda atau tidak. Pada hasil *thresholding*, hasil gambar akan diberikan sel guna proses ekstraksi.



Gambar 12. Contoh *bits extraction*

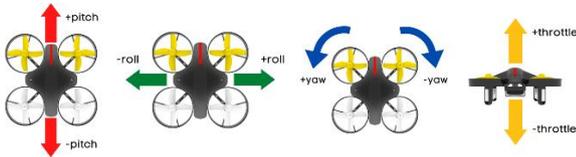
Pada tiap sel seperti yang ditunjukkan Gambar 12, warna hitam dan putih akan dihitung untuk menentukan nilai bit yang ditetapkan ke sel. Pada tahap ini telah dihasilkan nilai bit yang kemudian akan diidentifikasi sesuai dengan kamus ArUco *marker* yang digunakan pada program.

• *Marker Identification*

Tahap ini berisi dengan proses pemeriksaan kode tiap sel yang sebelumnya sudah diekstraksi disesuaikan dengan kamus yang sudah dideklarasikan dan akan menghasilkan id yang terdeteksi.

G. Konsep Gerak *Quadcopter* Terhadap Letak ArUco *Marker*

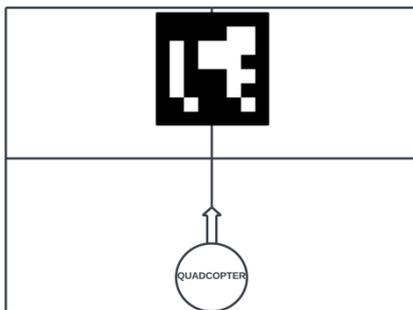
Quadcopter dapat terbang dengan beberapa kode perintah yang berbeda yakni *take-off*, *yaw*, dan *position* target. *Position* target akan membuat *quadcopter* dapat melakukan gerak *roll* dan *pitch* seperti yang ditunjukkan pada Gambar 13. Pada penelitian ini digunakan *library* DroneKit dikarenakan memiliki penulisan program yang sederhana dan lengkap.



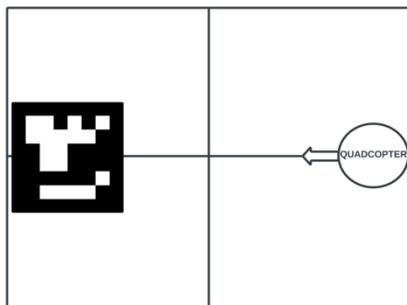
Gambar 13. Macam-macam gerakan *quadcopter*

Dengan menggunakan *function* “*send_body_ned_velocity*” yang didapat dari *library* DroneKit, terdapat 3 hal yang perlu diperhatikan yakni data X, Y, dan Z. Semua data ini dijadikan nilai ketika pemanggilan *function* di atas kemudian dimasukkan pada parameter yang ada di dalam *function* tersebut. Dalam *function* tersebut terdapat 3 parameter yang dapat diatur yakni “*velocity_x*” berperan untuk membuat *quadcopter* bergerak *pitch*, “*velocity_y*” berperan untuk membuat *quadcopter* bergerak *Roll*, dan “*velocity_z*” berperan untuk mengatur *throttle quadcopter*.

Letak objek deteksi pada Gambar 14 terletak di arah depan *quadcopter*. Hal ini mengharuskan *quadcopter* berjalan maju agar dapat memosisikan tubuhnya tepat di atas objek deteksi. Dikarenakan membutuhkan gerakan maju, maka parameter yang diatur yakni “*velocity_x*” untuk memberikan gerakan *pitch* pada *quadcopter*.



Gambar 14. Konsep *tracking* gerak maju



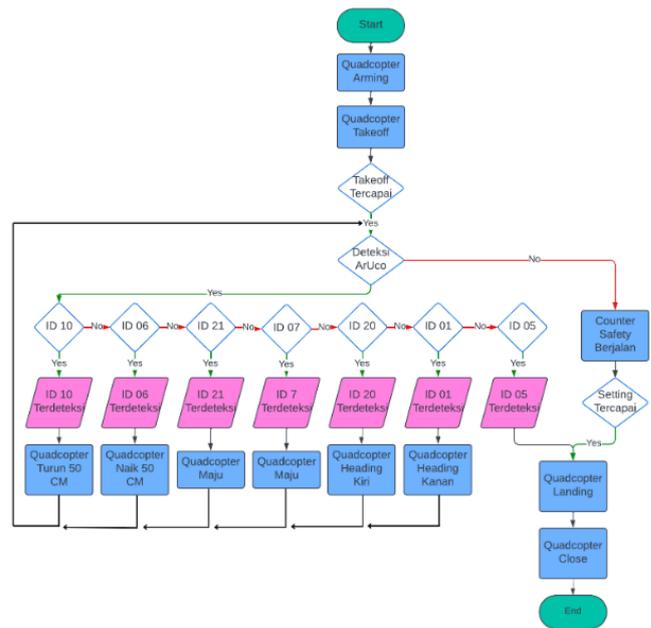
Gambar 15. Konsep *tracking* gerak ke kiri

Letak objek deteksi pada Gambar 15 terletak di arah sisi kiri *quadcopter*. Hal ini mengharuskan *quadcopter* berjalan ke kiri agar dapat memosisikan tubuhnya tepat di atas objek deteksi. Dikarenakan membutuhkan gerakan ke kiri, maka parameter yang diatur yakni “*velocity_y*” untuk memberikan gerakan *roll* pada *quadcopter*.

H. *Flowchart Misi Terbang Quadcopter*

Pada penelitian ini menggunakan 7 ArUco *marker* dengan ID yang berbeda. Masing-masing ArUco *marker* memiliki perintah tersendiri yang akan menggerakkan *quadcopter* sehingga dapat menyelesaikan misi. Pada penelitian ini juga ditanamkan sebuah fitur keamanan agar lebih aman dalam proses penelitian.

Selain pada gerakan yang diatur pada konsep *tracking* sebelumnya, dalam penelitian ini juga memungkinkan *quadcopter* dapat berbelok arah hingga menurunkan dan menaikkan ketinggian dari *quadcopter* itu sendiri. Hampir semua gerakan dasar *quadcopter* digunakan dalam penelitian agar didapatkan analisis yang lebih banyak dari tiap gerakannya.



Gambar 16. *Flowchart* misi *quadcopter*

Seperti yang ditunjukkan pada Gambar 16, sebelum membaca objek deteksi, *quadcopter* akan *arming* terlebih dahulu dan *take-off*. Setelah *take-off* di ketinggian yang ditentukan, kamera akan terbuka dan akan mulai membaca objek deteksi dalam hal ini adalah ArUco *marker*. Jika salah satu dari tujuh *marker* terbaca, maka *quadcopter* akan melakukan sesuai yang diperintahkan. Namun, jika tidak terbaca *marker*, *quadcopter* masuk dalam mode *safety* dan akan terus berjalan ke depan hingga *setting counter safety* tercapai yang kemudian akan *Landing* secara otomatis. Saat mendeteksi *marker*, *quadcopter* akan memosisikan dirinya yang membuat ArUco *marker* berada di tengah *frame*. Hal ini

bertujuan untuk menjaga *quadcopter* agar tetap berada di lintasan.

III. HASIL DAN PEMBAHASAN

Hasil pengembangan merupakan sebuah produk yang telah dihasilkan baik berupa *hardware* maupun *software* yang telah melewati proses perencanaan, identifikasi, analisis, dan evaluasi. Pengembangan *hardware* dan elektronik menghasilkan *quadcopter* dan objek deteksi seperti pada Gambar 17 yang sebelumnya sudah didesain. Komponen kelistrikan *quadcopter* telah terpasang pada *frame* F450 yang kemudian dapat dilakukan *flash firmware* dan pengujian.



Gambar 17. Hasil pengembangan

Objek ArUco *marker* dibuat dengan kertas asturo yang memiliki kepekatan warna hitam yang lebih gelap. Pemilihan kertas ini bertujuan untuk meningkatkan kualitas deteksi. Objek disusun dengan ukuran 40 cm x 40 cm dengan bidang alas berukuran 50 cm x 50 cm.

A. Uji Komponen Elektronik

Uji komponen elektronik dilakukan pada semua komponen. Dari pengujian didapatkan bahwa semua komponen mampu bekerja dalam keadaan baik kecuali telemetri dan *micro camera*. Berikut merupakan hasil pengujian dari 2 komponen tersebut.

1) *Telemetry*: Telemetri memiliki dua buah lampu indikator yakni lampu hijau dan merah. Lampu hijau mengindikasikan koneksi sedangkan lampu merah mengindikasikan pengiriman data.



Gambar 18. Status koneksi telemetri *ground*

Pada Gambar 18 merupakan telemetri *ground* yang menunjukkan lampu hijau menyala terus menerus. Terdapat 2

bagian telemetri yakni *Ground* dan *Air*. Hidupnya lampu hijau ini menandakan koneksi telemetri sudah berhasil.



Gambar 19. Pembacaan kualitas koneksi telemetri

Kualitas koneksi telemetri dapat dilihat yang ditunjukkan pada panah merah di Gambar 19. Semakin jauh jarak wahana dengan GCS, maka semakin buruk kualitas koneksi yang dimiliki. Tabel 3 merupakan data pengujian telemetri.

Tabel 3. Hasil uji kualitas koneksi telemetri

Jarak (meter)	Hasil
1	100%
2	100%
3	99%
4	98%
5	97%
6	36%
10	20%

Tabel 3 menunjukkan bahwa pada jarak 6 meter, kualitas koneksi telemetri sudah turun. Ini mengartikan bahwa jarak aman telemetri agar dapat berkomunikasi dengan baik di angka 5 meter.

Adapun persentase nilai koneksi telemetri dapat berubah-ubah sehingga menyebabkan pengiriman data tidak konsisten. Dengan nilai uji ini digunakan dalam pengembangan penelitian sehingga dapat berjalan sesuai dengan judul penelitian.

2) *Kamera*: Pengujian dilakukan pada siang hari yang memiliki intensitas cahaya maksimal. Pengujian kamera ditujukan untuk mempersiapkan prototipe dapat bergerak mengikuti perintah ArUco *marker*.



Gambar 20. Pengujian *micro camera*

Seperti yang ditunjukkan pada Gambar 20, pengujian dihasilkan bahwa kamera berjenis *micro camera* yang dipakai dalam penelitian ini tidak mampu membaca ArUco marker dengan baik sehingga dilakukan analisis lanjutan untuk mencari solusi dalam permasalahan tersebut.



Gambar 21. Pengujian kamera *smartphone*

Dari penggunaan kamera *smartphone* dan pemanfaatan aplikasi DroidCam, didapat hasil deteksi yang cukup baik seperti Gambar 21. Namun terdapat keterlambatan pengiriman video yang mengakibatkan sedikit terganggunya proses Bergeraknya *quadcopter* dalam mendeteksi ArUco marker. Dengan hal ini, digunakanlah kamera *smartphone* untuk menangkap video objek deteksi.

Pengujian komponen elektronik didasarkan untuk melihat hasil dari ketepatan kalibrasi yang sudah dilakukan sebelumnya. Komponen yang tidak terkalibrasi dengan baik akan berpengaruh besar pada jalannya penelitian. Masing-masing komponen memiliki peranan penting untuk membangun *quadcopter* bekerja dengan baik. Dari hasil uji memperlihatkan bahwa beberapa komponen terdapat kekurangan yang akan mempengaruhi hasil penelitian seperti telemetri dan kamera.

B. Uji Software

Pengujian *software* mencakup semua hal yang berkaitan dengan *firmware* yang dipakai pada *quadcopter*. Pengujian *software* bertujuan untuk melihat kesiapan berjalannya proses penelitian. berikut merupakan hasil pengujian *software* secara rinci.

1) *Uji Koneksi GCS*: Koneksi dengan GCS dilakukan menggunakan 2 cara yakni menggunakan kabel dan telemetri. Koneksi dengan GCS dapat dilakukan dengan menekan tombol *connect* pada tampilan awal *Mission Planner*. Indikator terkoneksi dengan GCS akan ditampilkan seperti Gambar 22. Pada tampilan *home Mission Planner* terdapat beberapa informasi seperti posisi *accel* hingga *altitude quadcopter*.



Gambar 22. Pengujian koneksi GCS

2) *Uji Deteksi ArUco Marker*: Pengujian ditujukan untuk melihat kesiapan kamera terhadap jarak deteksi. Dengan hal ini, jarak batas aman deteksi yakni 400 cm. Jika *quadcopter* di bawah batas aman, maka *quadcopter* dapat mendeteksi ArUco marker dan berjalan sesuai dengan perintah yang diberikan hingga jarak 50 cm dari objek deteksi seperti hasil pengujian yang ditunjukkan pada Tabel 4 berikut.

Tabel 4. Hasil uji deteksi objek (ArUco marker)

No.	Jarak Kamera (cm)	Hasil
1	50	Tidak Terdeteksi ID
2	100	Terdeteksi ID
3	150	Terdeteksi ID
4	200	Terdeteksi ID
5	250	Terdeteksi ID
6	300	Terdeteksi ID
7	350	Terdeteksi ID
8	400	Terdeteksi ID

C. Hasil Tracking Quadcopter Terhadap Letak ArUco Marker

Jika digabungkan kedua konsep yakni gerakan *pitch* dan *roll* seperti yang sudah dipaparkan pada bagian metodologi, maka bisa didapatkan hasil seperti Gambar 23. Terlihat mulai dari gambar 1 ke gambar 4, posisi objek deteksi berjalan menuju ke sisi tengah. Ini mengartikan bahwa *quadcopter* dapat memosisikan badannya tepat di atas objek deteksi.



Gambar 23. Hasil tracking *quadcopter*

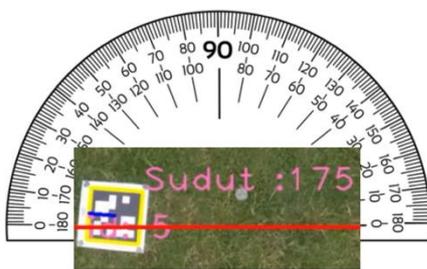


Gambar 24. Grafik koordinat X dan Y *quadcopter*

Jika *quadcopter* mampu *tracking* objek dan mengkoreksi posisinya, maka didapatkan grafik seperti Gambar 24. terlihat bahwa kedua titik yakni X dan Y yang mewakili posisi *quadcopter* cenderung mendekati titik 5. Nilai 5 ini merupakan titik tengah yang dimiliki oleh objek deteksi. Dengan ini, *quadcopter* sudah siap untuk berjalan sesuai misi dengan perintah yang diberikan oleh ArUco *marker*.

D. Uji Misi

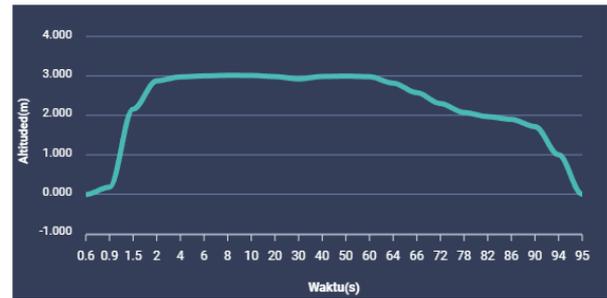
Pengujian dilakukan dengan beberapa variasi penempatan ArUco *marker* untuk membandingkan hasil antar lintasan. Dikarenakan jenis *quadcopter* yang dipakai adalah tipe *outdoor*, maka terdapat gangguan dari luar seperti angin yang akan mempengaruhi hasil penelitian. Angin yang berhembus kencang dan mengenai badan *quadcopter* akan sangat berpengaruh terhadap posisi dan *heading quadcopter*. Maka dari itu, salah satu parameter yang dianalisis adalah terkait dengan *heading quadcopter*.



Gambar 25. Pembacaan sudut dengan garis sumbu

Gambar 25 merupakan acuan pembacaan garis sumbu pada *frame* dan *heading quadcopter*. Garis sumbu merah merupakan sumbu *frame* dan garis sumbu biru merupakan garis *heading quadcopter*. *Heading* dijadikan sebagai parameter analisis dikarenakan dalam penerapan di dalam gudang diperlukan *quadcopter* yang dalam kondisi normal. Pengaruh dapat terjadi karena faktor eksternal seperti angin maupun faktor internal seperti komponen terutama sensor yang dalam hal ini adalah kompas.

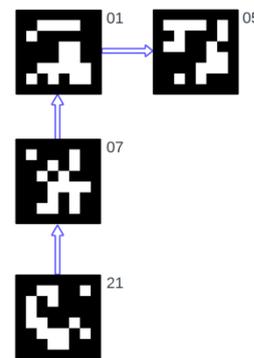
Dalam gambar tersebut terlihat bahwa garis sumbu berwarna biru lebih dominan mengarah pada sudut 175°. Pada gambar tersebut juga terlihat hasil deteksi sudut yang menghasilkan sudut 175°. Dengan hal ini, maka deteksi sudut yang dihasilkan dari penelitian masih dalam batas aman.



Gambar 26. Grafik *altitude quadcopter*

Persiapan uji data selain sudut ialah stabilitas *altitude* yang dimiliki oleh *quadcopter*. Dalam menjalankan misi, stabilitas nilai *altitude* menjadi salah satu kunci keberhasilan. Pada grafik Gambar 26 menunjukkan bahwa setelah *take-off* yakni pada ketinggian 0 sampai 3 meter, *quadcopter* memiliki nilai *altitude* yang cukup stabil. Pada titik 60s, *Quadcopter* memulai untuk *landing*. Dari titik tersebut, grafik sudah menurun dengan stabil sembari *quadcopter* mengoreksi posisinya. Adanya koreksi posisi terlihat pada titik 78s yang memiliki nilai *altitude* cenderung landai. Ketika *quadcopter* sudah memosisikan dan masuk dalam *threshold*, *quadcopter* akan melanjutkan proses *landing* hingga pada ketinggian 0 meter.

1) *Lintasan Ke-1*: Pada percobaan pertama dengan menggunakan lintasan pada Gambar 27, *quadcopter* dapat melewati seluruh ArUco *marker* hingga *landing* seperti yang ditunjukkan pada Gambar 28. Pada setiap *marker* yang dilewati dan ditunjukkan hasil pengujian pada Tabel 5, posisi *heading quadcopter* memiliki *error* sudut di angka 1°-11°.



Gambar 27. Lintasan ke-1

Error sudut ini dapat terjadi dikarenakan gangguan eksternal di lapangan yakni angin. Hal ini ditegaskan dengan tidak konsistennya *heading quadcopter* pada tiap terdeteksinya ArUco *marker*. Terdapat *error* sudut yang lebih besar pada saat *landing* yakni -11°. *Error* sudut yang besar ini mempengaruhi kualitas *landing quadcopter* yang hanya menyentuhkan 1 kaki *quadcopter* pada area *marker* seperti yang ditunjukkan pada Gambar 28.

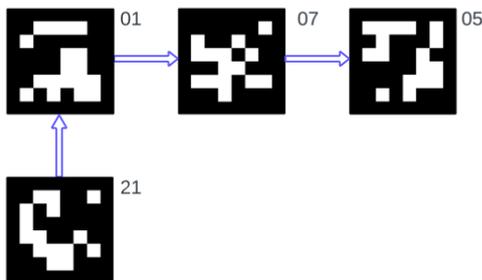


Gambar 28. Hasil landing percobaan ke-1

Tabel 5. Hasil pembacaan sudut percobaan ke-1

ID	Gerakan Yang Diminta	Sudut
21	Maju	+2°
07	Maju	+1°
01	Belok Kanan	-8°
05	Landing	-11°
Rata-rata Error Sudut		5,5°

2) *Lintasan Ke-2:* Pada percobaan kedua dengan menggunakan lintasan pada Gambar 29, *quadcopter* dapat melewati seluruh ArUco marker hingga landing seperti yang ditunjukkan pada Gambar 30. Pada setiap marker yang dilewati dan ditunjukkan hasil pengujian pada Tabel 6, posisi heading *quadcopter* memiliki error sudut di angka 1°-10°.



Gambar 29. Lintasan ke-2



Gambar 30. Hasil landing percobaan ke-2

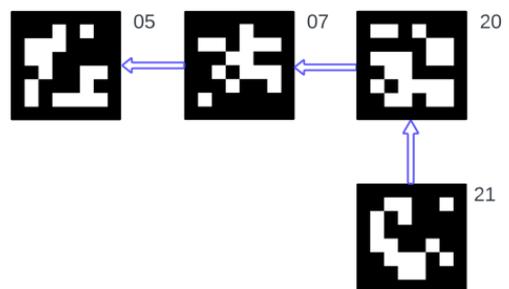
Tabel 6. Hasil pembacaan sudut percobaan ke-2

ID	Gerakan Yang Diminta	Sudut
21	Maju	+1°
01	Belok Kanan	-5°
07	Maju	-11°
05	Landing	-10°
Rata-rata Error Sudut		6,75°

Error sudut ini dapat terjadi dikarenakan gangguan eksternal di lapangan yakni angin. Hal ini ditegaskan dengan tidak konsistennya heading *quadcopter* pada tiap terdeteksinya ArUco marker. Terdapat *error* sudut yang lebih besar pada saat landing yakni -10°. Seperti yang ditunjukkan pada Gambar 30, walaupun dengan sudut yang masih besar, lebih dari setengah badan *quadcopter* dapat masuk ke dalam ArUco marker. Hasil landing pada percobaan kedua lebih baik daripada percobaan pertama. Hal ini dapat terjadi dikarenakan adanya tekanan angin, pengiriman data video, dan pengiriman data telemetri yang kurang baik.

3) *Lintasan Ke-3:* Pada percobaan ketiga dengan menggunakan lintasan pada Gambar 31, *quadcopter* dapat melewati seluruh ArUco marker hingga landing seperti yang ditunjukkan pada Gambar 32. Pada setiap marker yang dilewati dan ditunjukkan hasil pengujian pada Tabel 7, posisi heading *quadcopter* memiliki error sudut di angka 0°-8°.

Error sudut ini dapat terjadi dikarenakan gangguan eksternal di lapangan yakni angin. Hal ini ditegaskan dengan tidak konsistennya heading *quadcopter* pada tiap terdeteksinya ArUco marker. Seperti yang ditunjukkan pada Gambar 32, walaupun dengan sudut yang masih besar, lebih dari setengah badan *quadcopter* dapat masuk ke dalam ArUco marker. Hasil landing pada percobaan ketiga hampir serupa dengan percobaan kedua. Hal ini dapat terjadi dikarenakan adanya tekanan angin, pengiriman data video, dan pengiriman data telemetri yang kurang baik. Namun, pada percobaan ketiga memiliki *error* sudut yang lebih kecil yakni tidak lebih dari 8°.



Gambar 31. Lintasan ke-3

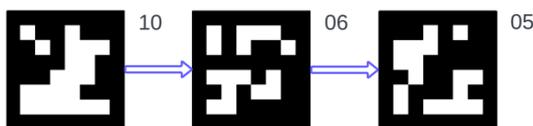


Gambar 32. Hasil landing percobaan ke-3

Tabel 7. Hasil pembacaan sudut percobaan ke-3

ID	Gerakan Yang Diminta	Sudut
7	Maju	0(180°)
20	Belok Kiri	+7°
21	Maju	+8°
05	Landing	+7°
Rata-rata Error Sudut		5,5°

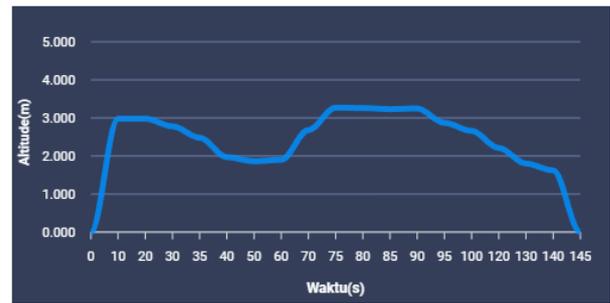
4) *Lintasan Ke-4*: Sesuai dengan flowchart, ArUco marker dengan ID 10 memberi perintah turun 50 cm dan ID 06 memberi perintah Naik 50 cm dari ketinggian *take-off* yakni 3 meter. Hasil percobaan ketiga dengan menggunakan lintasan pada Gambar 33 dapat terlihat pada grafik Gambar 35, bahwa pada saat *quadcopter* berjalan di waktu 10s-40s *quadcopter* menurunkan ketinggian di angka 2 meter, pada waktu 60 s – 75 s *quadcopter* menaikkan ketinggian di angka 3,2 meter dan mulai proses *landing* pada waktu 90s. Adapun perbedaan hasil capaian *altitude* yang diperintahkan dari ArUco marker dikarenakan kualitas transmisi data yang dimiliki oleh telemetri yang didasarkan dari hasil pengujian telemetri. Dalam percobaan ini *quadcopter* dapat *landing* seperti yang ditunjukkan pada Gambar 34.



Gambar 33. Lintasan ke-4



Gambar 34. Hasil landing percobaan ke-4



Gambar 35. *Altitude quadcopter* percobaan ke-4

Tabel 8. Hasil pembacaan sudut percobaan ke-4

ID	Gerakan Yang Diminta	Sudut
10	Turun	-1°
06	Naik	-4°
05	Landing	-3°
Rata-rata Error Sudut		2,6°

Pada percobaan keempat, *quadcopter* dapat melewati seluruh ArUco marker hingga *landing*. Pada setiap marker yang dilewati dan ditunjukkan hasil pengujian pada Tabel 8, posisi heading *quadcopter* memiliki *error* sudut di angka -1° hingga -4°.

Error sudut ini dapat terjadi dikarenakan gangguan eksternal di lapangan yakni angin. Hal ini ditegaskan dengan tidak konsistennya heading *quadcopter* pada tiap terdeteksinya ArUco marker. Seperti yang ditunjukkan pada Gambar 34, walaupun dengan *error* sudut yang cukup besar, lebih dari setengah badan *quadcopter* dapat masuk ke dalam ArUco marker. Hal ini dapat terjadi dikarenakan adanya tekanan angin, pengiriman data video, dan pengiriman data telemetri yang kurang baik. Namun, pada percobaan keempat memiliki rata-rata *error* sudut yang lebih kecil yakni di angka 2,6°.

IV. SIMPULAN

Dengan penggunaan kamera dari *smartphone* yang dikarenakan kualitas pembacaan *micro camera* yang kurang baik, didapat *error* sudut terbesar adalah 6,75°. *Error* sudut ini dapat dipengaruhi dari dorongan angin. Dengan *error* sudut tersebut, diperlukan pemrograman untuk koreksi sudut *quadcopter*, sehingga didapat kualitas navigasi terbang *quadcopter* yang menggunakan ArUco marker secara maksimal.

DAFTAR PUSTAKA

- [1] O. Pahlevi, A. Mulyani, dan M. Khoir, "Sistem informasi inventori barang menggunakan metode object oriented di PT. Livaza Teknologi Indonesia Jakarta," *J. Prosisko*, vol. 5, no. 1, pp. 27–35, 2018.
- [2] D. Gill, "CEVA Logistics Chosen to Operate Warehouse for IKEA in Mexico," *Logistics Manager*, 3 Agustus 2020. [Online]. Tersedia: <https://logistics-manager.com/ceva-logistics-chosen-to-operate-warehouse-for-ikea-in-mexico/>.
- [3] H. M. Kandeel, E. Abdelmaksod, dan A. Elnady, "Modeling and control of x-shape quadcopter," *IOSR Journal of Mechanical and*

- Civil Engineering*, vol. 19, pp. 46–57, 2022.
- [4] O. Tatale, N. Anekar, S. Phatak, dan S. Sarkale, "Quadcopter: design, construction and testing," *International Journal for Research in Engineering Application & Management*, vol. 4, Special Issue AMET-2018, pp. 1–7, 2018.
- [5] F. N. Sekarsih, F. Nucifera, R. Pringgondani, dan A. L. Permatasari, "Pengenalan citra drone sebagai data geo-spasial untuk Sekolah Menengah Atas," *Jurnal Pengabdian Kepada Masyarakat Nusantara*, vol. 5, no. 1, pp. 250–258, 2024.
- [6] ArduPilot, "What is ArduPilot?," 2024. [Online]. Available: <https://ardupilot.org/>. [Diakses: 12-Jun-2024].
- [7] A. K. Rahman, H. Supriyanto, dan T. Meizinta, "Rancang bangun dan implementasi sistem kendali quadcopter melalui jaringan internet berbasis lokasi menggunakan smartphone android," in *Prosiding Seminar Nasional Instrumentasi, Kontrol dan Otomasi*, 2018, pp. 307–318.
- [8] R. I. Hartono, A. Rusdinar, dan W. A. Cahyadi, "Sistem pengenalan tanda buatan untuk navigasi drone balon udara," *eProceedings of Engineering*, vol. 9, no. 5, pp. 2517–2520, 2022.
- [9] A. Haniffullah, I. A. Bangsa, dan A. Stefanie, "Implementasi quadcopter pembawa arm robot pembersih kaca jendela gedung bertingkat," *Edu Elekrika Journal*, vol. 9, no. 2, pp. 36–42, 2020.
- [10] H. Muchtar dan R. Apriadi, "Implementasi pengenalan wajah pada sistem penguncian rumah dengan metode template matching menggunakan Open Source Computer Vision Library (OpenCV)," *RESISTOR (elektRONika kEndali telekomunikaSI tenaga liSTRik kOmputeR)*, vol. 2, no. 1, pp. 39–42, 2019.
- [11] V. Wiley dan T. Lucas, "Computer vision and image processing: A paper review," *International Journal of Artificial Intelligence Research*, vol. 2, no. 1, pp. 28–36, 2018.
- [12] N. F. Ardiansyah, A. Rabiâ, D. Minggu, dan W. Dirgantara, "Computer vision untuk pengenalan obyek pada peluncuran roket kendaraan tempur," *JASIEK (Jurnal Aplikasi Sains, Informasi, Elektronika Dan Komputer)*, vol. 1, no. 1, pp. 28–37, 2019.
- [13] Pemrograman Matlab, "Pengolahan Citra Digital, Pengolahan Video, Pengenalan Pola, dan Data Mining," 2020. [Online]. Available: <https://pemrogramanmatlab.com/pengenalan-pola-citra-digital-menggunakan-matlab/>. [Diakses: 12-Jun-2024].
- [14] A. Triwahyudin, H. K. Safitri, dan M. Fauziyah, "Pembacaan jarak dan kecepatan dengan Aruco marker pada sistem koper follow me beroda," *Majalah Ilmiah Teknologi Elektro*, vol. 21, no. 1, pp. 97–106, 2022.
- [15] B. Li, J. Wu, X. Tan, dan B. Wang, "ArUco marker detection under occlusion using convolutional neural network," in *2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE)*, Dalian, China, 2020, pp. 706–711, doi: 10.1109/CACRE50138.2020.9230250.
- [16] W. Suparta, A. C. Ahmad, dan A. H. A. Tharim, "Implementation of hexacopter for package delivery," *Majlesi Journal of Electrical Engineering*, vol. 18, no. 1, pp. 205–215, 2024.
- [17] A. S. Priambodo, F. Arifin, A. Nasuha, A. Muslikhin, dan A. Winursito, "A vision and GPS based system for autonomous precision vertical landing of UAV quadcopter," *Journal of Physics: Conference Series*, vol. 2406, no. 1, 2022.
- [18] M. F. Ahmed, M. N. Zafar, dan J. C. Mohanta, "Modeling and analysis of quadcopter F450 frame," in *2020 International Conference on Contemporary Computing and Applications (IC3A)*, 2020, pp. 196–201.
- [19] A. Faiz, A. Kusyanti, dan R. A. Siregar, "Implementasi pengamanan protokol MAVLink menggunakan algoritma kriptografi Simeck," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 10, pp. 4678–4685, 2022.
- [20] A. A. Pasanca, M. Munadi, dan M. Muchammad, "Perancangan quadcopter F450 dengan flight controller Pixhawk 2.4.8 sebagai alat deteksi keretakan," *Jurnal Teknik Mesin*, vol. 11, no. 3, pp. 267–272, 2023.
- [21] H. M. Qays, B. A. Jumaa, dan A. D. Salman, "Design and implementation of autonomous quadcopter using SITL simulator," *Iraqi Journal of Computer, Communication, Control and System Engineering*, vol. 20, no. 1, pp. 1–16, May 2020, doi: 10.33103/uot.ijecce.20.1.1.
- [22] X. Chen, "Introduction and analysis of Python software," *FCIS*, vol. 5, no. 2, pp. 41–43, Sep. 2023, doi: 10.54097/fcis.v5i2.12348.
- [23] M. Košuda, P. Lipovsky, Z. Szöke, M. Fil'ko, J. Novotňak, dan F. Heško, "MAVLink messaging protocol as potential candidate for the UTM communication," in *2020 New Trends in Signal Processing (NTSP)*, Demanovska dolina, Slovakia, 2020, pp. 1–7, doi: 10.1109/NTSP49686.2020.9229550.
- [24] E. H. Houssein, M. M. Emam, dan A. A. Ali, "An efficient multilevel thresholding segmentation method for thermography breast cancer imaging based on improved chimp optimization algorithm," *Expert Systems with Applications*, vol. 185, 115651, 2021.