

Implementasi *Quadcopter* Pembawa *Arm Robot* Pembersih Kaca Jendela Gedung Bertingkat

Al Haniffullah¹, Insani Abdi Bangsa², dan Arnisa Stefanie³

^{1,2,3}Jurusian Teknik Elektro, Fakultas Teknik, Universitas Singaperbangsa Karawang
Jl. HS.Ronggowaluyo,Telukjambe Timur, Karawang, Jawa Barat, 41361, Indonesia

al.haniffullah16016@student.unsika.ac.id¹, iabdi.bangsa@ft.unsika.ac.id², arnisa.stefanie@staff.unsika.ac.id³

Abstrak— *Quadcopter* adalah jenis robot *drone multirotor* yang menggunakan empat buah motor penggerak yang dikendalikan dengan menggunakan *Flight Controller*. *Quadcopter* banyak digunakan dalam berbagai kebutuhan, seperti pada penelitian, eksplorasi, pengambilan citra, juga pengamatan. Pada penelitian ini dilakukan implementasi *quadcopter* yang mampu terbang dengan stabil dan dapat mempertahankan ketinggian dan posisinya. Seluruh proses pengendalian dan kestabilan *quadcopter* diatur menggunakan *PID controller* dan diproses oleh perangkat APM 2.8 sebagai *flight controller*, yang dilengkapi dengan sensor *gyroscope*, *accelerometer* dan modul *GPS*. Hasil dari implementasi dan penelitian ini adalah : 1. Menghasilkan *quadcopter* tipe X berukuran 550mm dengan bobot 1,5 Kg dan menggunakan 4 buah *propeller* dengan motor *brushless* 980KV; 2. *Quadcopter* dapat mempertahankan ketinggian terbangnya menggunakan nilai $P = 1,5$ dengan *rise time* 0,5 detik dan nilai rata-rata *steady state error* sebesar 0,20 meter; 3. Untuk sistem kestabilan gerak *pitch* dan *roll* menggunakan nilai $P=0,25$, $I = 0,15$ dan $D = 0,04$.

Kata kunci— *Quadcopter*, *Flight controller*, *Ardupilot Mega*, Sistem kendali *PID*.

The quadcopter is a multirotor drone that uses four drivester-controlled motors using flight controllers. The quadcopter is widely used in needs, such as in research, exploration, image retrieval, as well as observation. The study has been followed by a quadcopter implementation capable of steady flight and of maintaining altitude and position. The quadcopter's entire process of control and stability is controlled by a pid controller and is processed by the apm 2.8 device as flight controller, which is equipped with a gyroscope sensor, an accelerometer and a GPS module. The results of implementation and research are: 1. The quadcopter of 550mm x with a weight of 1.5 kg (5 lb) and uses 4 kilos on a brushless 980kv bike; 2. The quadcopter can maintain its altitude of altitude using a KP = 1.5 with a rise of time 0.5 seconds and a steady state average of 0.06 meters; 3. For a pitch and roll pitch system using KP value = 0.25, ki = 0.15 and kd = 0.04

Keywords— *Quadcopter*, *Flight controller*, *Ardupilot Mega*, *PID System control*.

I. PENDAHULUAN

Perkembangan teknologi ditujukan untuk membantu manusia dalam melakukan pekerjaan tertentu agar lebih mudah dan efisien, misalnya pekerjaan yang membutuhkan alat berat dan memiliki kemungkinan besar terjadi kecelakaan kerja. Salah satu pekerjaan yang paling beresiko adalah pembersihan kaca jendela gedung bertingkat. Untuk mencapai ketinggian gedung diperlukan seorang pekerja yang naik menggunakan sebuah gondola, atau cara lain yang dilakukan adalah dengan teknik *wall climbing* menuruni gedung dari tingkat paling atas. Pekerjaan tersebut dilakukan oleh manusia secara manual dengan mengandalkan perlengkapan *safety*. Sebagai contoh kecelakaan kerja pada pesawat gondola di Indonesia yaitu pada tanggal 3 September 2008 di lantai 16 blok D Apartemen Majesty Bandung akibat satu crane penyangga patah dan tali bajanya putus. Contoh lainnya lima orang pekerja tewas akibat terjatuh dari gondola pada ketinggian sekitar 130 meter saat melakukan pekerjaan di Tower RCTI, tanggal 28 Agustus 2008, kecelakaan tersebut diduga terjadi akibat tali sling putus.

Quadcopter atau yang biasa disebut *Drone* adalah salah satu teknologi berupa robot yang dapat terbang di udara bebas diudara, sehingga dapat dipergunakan untuk melakukan pekerjaan yang tidak dapat dijangkau oleh manusia seperti untuk *monitoring* kemacetan jalan, survei dan pemetaan, robot mata-mata, *monitoring* bencana alam, *monitoring* kebakaran

lahan, alat bantu pemadam kebakaran, media pencarian pada misi *Search and Rescue* (SAR).

Dilihat dari pemanfaatan *Quadcopter* dan permasalahan pekerja pembersih kaca gedung bertingkat, maka dilakukan implementasi *Quadcopter* yang dapat membawa robot *manipulator* berupa *Arm Robot*, dengan kemampuan menstabilkan terbang, ketinggian dan posisinya, agar *Arm Robot* dapat melakukan pekerjaanya dengan baik.

Dengan dilakukan penelitian tersebut diharapkan dapat memberikan inovasi teknologi baru, mengurangi risiko terjadinya kecelakaan kerja dan meningkatkan efisiensi waktu dan biaya dalam proses perawatan Gedung bertingkat.

II. KAJIAN PUSTAKA

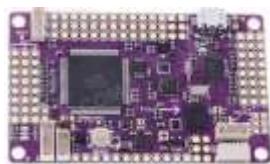
A. *Quadcopter*

Quadcopter merupakan salah satu jenis pesawat tanpa awak atau yang biasa disebut *Drone*. Pesawat tanpa awak dapat dikendalikan dari jarak jauh oleh penggunanya, menggunakan *radio remote control* dan biasanya memiliki ukuran yang lebih kecil dari pada pesawat pada umumnya, sehingga *drone* banyak dimanfaatkan untuk mempermudah pekerjaan manusia yang berhubungan dengan ketinggian. *Quadcopter* memiliki 4 buah motor yang bekerja dengan memanfaatkan perubahan kecepatan untuk mengubah arah dan gerakan pada *quadcopter*. Gerakan yang dapat dilakukan oleh *quadcopter* adalah gerakan *throttle*, *roll*, *pitch* dan *yaw*.

B. Flight Control Ardupilot Mega 2.8

Ardupilot adalah perangkat keras *flight controller* yang berfungsi untuk mengatur seluruh sistem pada *quadcopter* dan juga dapat digunakan untuk proyek *autopilot*. Ardupilot berbasis *platform open-source* menggunakan *mikrokontroler* Arduino Mega, yang dikembangkan oleh Chris Anderson dan Jordi Munoz dari DIY Drones. Modul ini berbasis *open-source* paling berkembang untuk modul autopilot. Baik *autopilot* untuk pesawat (*Arduplane*), *Multicopter* (*Arducopter*) dan juga kendaraan darat (*Ardurover*).

Ardupilot Mega memiliki *main processor board*, yang menggunakan prosesor ATMega2560 dan ATMega328 sebagai pendukung prosesor yang digunakan sebagai *interface* RC. Bentuk fisik *board* APM ditunjukkan pada Gambar 1 dibawah ini.



Gambar 1. Bentuk fisik APM

Board APM dilengkapi dengan *input* dan *output* pin yang dapat digunakan untuk koneksi antara APM dengan komputer, *power module*, *radio receiver*, dan perangkat penyusun *drone* lainnya. Ardupilot Mega dilengkapi dengan beragam sensor yang disusun menjadi satu kesatuan bernama *IMU (Inertial Measurement Unit)*. Sensor tersebut ditanamkan pada *shield board* APM yang terletak dibagian paling atas *board*, sensor-sensor tersebut antara lain adalah :

- a. *Gyroscope*
- b. *Accelerometer*
- c. *Sensor Kompas/Magnetometer*
- d. *Barometer*

C. Mission Planner

ArduPilot Mega membutuhkan *software Mission planner* untuk merencanakan misi, atau mengunggah *firmware* baru. Pada Gambar 2 adalah tampilan dari *software mission planer* yang di gunakan untuk menghubungkan antara laptop dengan Ardupilot Mega.



Gambar 2. Tampilan *Software Mission Planner*

Software mission planner dapat memantau semua status dari pesawat baik ketinggian, jalur terbang, status baterai, dan lainnya. Dengan penambahan Kit Telemetri, kita dapat melacak UAV Anda secara *real time*, atau bahkan mengubah misi Anda saat *ArduPilot Mega* UAV sedang berada di udara.

D. Modul GPS (Global Position System) NEO-M8

Modul GPS mengukur lokasi dari drone dengan mengukur seberapa lama sinyal bergerak dari satelit. Modul ini dapat juga digunakan untuk memperkirakan ketinggian dan posisi *quadcopter*. Akurasi dari GPS adalah sekitar 5 meter. Fitur utama dari modul GPS adalah menerbangkan drone melalui *way-point* yang sudah ditetapkan secara otomatis.

Penggunaan GPS juga dapat dimanfaatkan untuk mengaktifkan fitur *Position Hold* dan *Altitude Hold* yang memungkinkan *drone* dapat terbang secara stabil pada posisi yang telah ditetapkan.



Gambar 3. GPS NEO-M8

Modul GPS yang dapat digunakan untuk proyek *quadcopter* salah satunya adalah GPS NEO-M8 yang memiliki bentuk seperti pada Gambar 3.

E. ESC (Electronic Speed Control)

PWM atau *pulse width modulation* adalah metoda untuk mengatur kecepatan perputaran motor. Sinyal PWM tersebut kemudian dikirimkan ke komponen ESC (*Electronic Speed Control*) yang berfungsi untuk menerjemahkan sinyal dari *flight controller* untuk menambah atau mengurangi daya yang diberikan ke *brushless motor* drone. Tipe ESC 40A ditunjukkan pada Gambar 4 di bawah ini.



Gambar 4. Bentuk fisik ESC

F. Brushless Motor

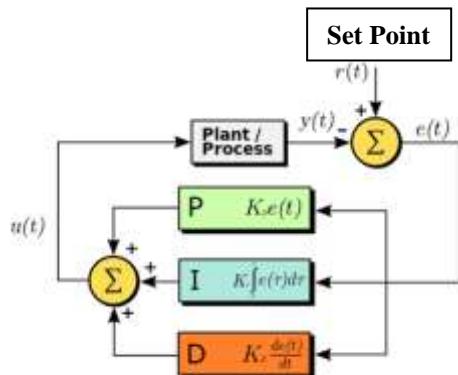
Motor brushless adalah motor yg umumnya memiliki tiga buah *coil* yang masing-masing memiliki satu kabel untuk masuk ke ESC atau baterai, sehingga jika dilihat dari luar *brushless motor* selalu memiliki tiga buah kabel. Di *motor brushless* ini, yang berputar bukanlah bagian *coil*-nya, karena *coil brushless* menempel di *chasing* motornya. Bentuk fisik motor *brushless* ditunjukkan pada Gambar 5.



Gambar 5. Bentuk Fisik *SunnySky brushless motor*

G. PID Controller

PID merupakan sistem kontrol atau kendali yang biasa digunakan untuk menentukan presisi suatu instrumentasi agar dapat menghasilkan *output* yang tidak jauh berbeda dari nilai yang diinginkan. Kontroler PID terdiri dari Proposional, Integral, dan Derivatif yang ketiganya saling menutupi dan melengkapi kelebihan dan kekurangan dari masing-masing kontroler apabila digabungkan secara paralel. Blok diagram kontroler PID ditunjukkan pada Gambar 6.



Gambar 6. Blok diagram PID

Pengontrol PID membandingkan nilai *set point* dengan nilai hasil pembacaan *error* dari *plant/process*, kemudian dengan menggunakan nilai PID yang tepat, maka kontroler akan memberikan aksi pada sistem tersebut untuk menghasilkan *output* yang nilainya tidak jauh atau bahkan sesuai dengan nilai *set point* yang diberikan. Proses pengontrolan PID akan terus berlanjut selama sistem masih dijalankan, karena PID merupakan sistem kontrol dengan loop tertutup.

Proses pencarian nilai PID disebut dengan proses *tuning*, cara untuk melakukan *tuning* PID secara manual adalah dengan melihat tabel parameter PID dan kemudian menganalisa respon yang dihasilkan dari nilai yang dimasukan ke parameter PID. Tabel Parameter PID ditunjukkan pada Tabel 1.

TABEL I. PARAMETER PID

Parameter	Rise Time	Over Shoot	Settling Time	S-S Error
K _p	Berkurang	Bertambah	Minor Change	Berkurang
K _i	Berkurang	Bertambah	Bertambah	Menghilang
T _d	Minor Change	Berkurang	Berkurang	Minor Change

Respon yang dari pemberian nilai pada parameter PID dapat dianalisa dengan melihat karakter dari sistem yang dikontrol. Apabila karakter sistem tersebut belum sesuai keinginan atau masih jauh dari kondisi stabil, maka perlu dilakukan proses *tuning* dengan mengurangi atau menambahkan konstanta kendali yang mempengaruhi karakter yang menonjol dari sistem tersebut.

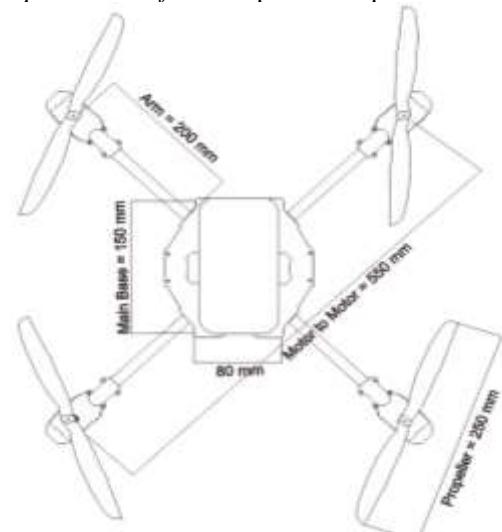
III. METODOLOGI PENITIANS

Metode penelitian yang dilakukan pada penelitian ini dilakukan dengan tahap-tahap sebagai berikut :

A. Desain dan Perancangan Sistem Quadcopter

1) Desain Kerangka Quadcopter

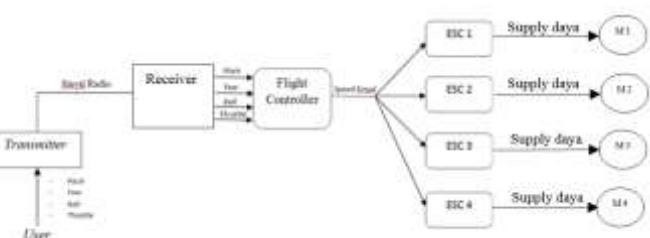
Perancangan kerangka/*frame quadcopter* dimulai dengan membuat desain 2 dimensi menggunakan *software AutoCad*. Rangka *quadcopter* yang digunakan memiliki ukuran 550 mm yang dihitung dari jarak antar motor. *Frame* dibuat menggunakan bahan dasar alumunium dan papan PVC untuk bagian *main basenya*. Bahan-bahan tersebut memiliki bobot yang ringan, sehingga cocok untuk mengurangi beban angkat dari *quadcopter*. Desain *frame* dapat dilihat pada Gambar 7.



Gambar 7. Desain frame quadcopter

2) Rangkaian Komponen Sistem

Perancangan komponen adalah tahapan penyusunan komponen pembentuk sistem *quadcopter*. Komponen-komponen tersebut dihubungkan sesuai dengan blok diagram dari sistem kerja *quadcopter* yang ditunjukkan pada Gambar 8.

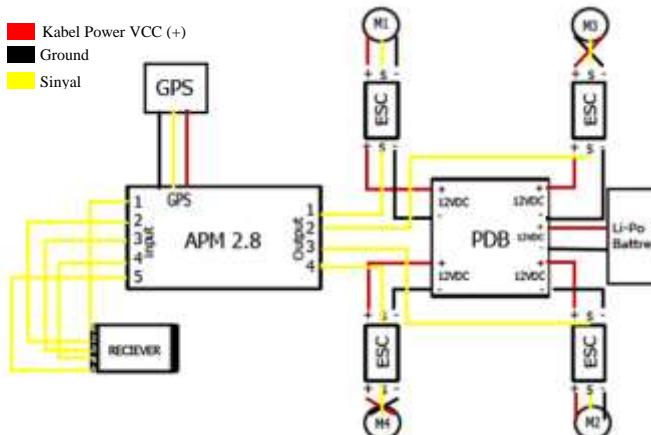


Gambar 8. Diagram blok sistem kerja quadcopter

Quadcopter dikendalikan menggunakan sistem *radio control* dengan *remote control* yang memiliki 4 *channel* mode gerak yaitu, *pitch* (Maju dan mundur), *Roll* (kanan dan kiri), *Yaw* (berputar pada poros), *Throttle* (terbang naik dan turun). *Transmitter remote control* akan mengirimkan *input* berupa frekuensi sinyal radio dari tiap *channel* ke *receiver* yang ada pada drone, kemudian *receiver* melakukan pengiriman sinyal tersebut dengan masing-masing *channel* yang memiliki media transmisi ke *flight controller*.

Sinyal radio yang diterima oleh *receiver* akan diproses oleh *flight controller* menjadi *output* berupa sinyal kecepatan PWM (*pulse with modulation*). Sinyal PWM kemudian dikirimkan ke komponen *ESC* (*electronic speed controller*) sebagai input, komponen *ESC* tersebut berfungsi untuk mengatur kecepatan putaran motor drone dengan memberikan *input* tegangan sesuai dengan sinyal PWM yang diterima dari *flight controller*.

Sesuai dengan blok diagram kerja *quadcopter* pada Gambar 8, maka dibuat skema rangkaian untuk menghubungkan seluruh komponen seperti pada Gambar 9 agar dapat bekerja sesuai dengan sistem kerja yang telah dijelaskan.



Gambar 9. Wiring Diagram *quadcopter*

B. Perancangan Sistem Kendali PID

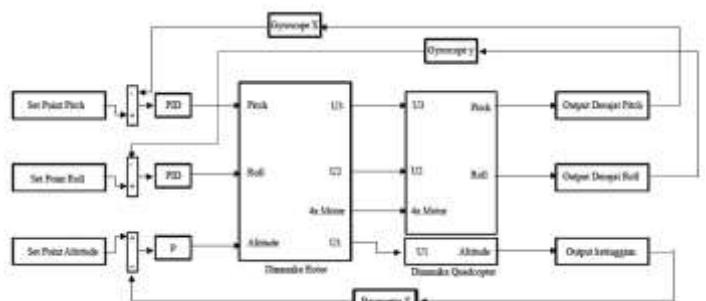
Sistem kendali *quadcopter* secara keseluruhan dikendalikan oleh *flight controller* dengan menerima *input* sinyal dari *radio receiver* yang terhubung dengan *user*. *Flight control* dilengkapi sistem kestabilan (*steady*) dengan menggunakan sensor *gyroscope* dan *accelerometer*. Sistem tersebut dapat bekerja menggunakan metode sistem kendali PID (*Proportional–Integral–Derivative*) yang diatur di dalam program *flight controller*. Cara kerja dari sistem kendali ditunjukkan pada Gambar 10.

Sensor *gyroscope* akan membaca perubahan orientasi gerak drone, dan *accelerometer* akan membaca percepatan putaran motor drone. Nilai yang dibaca kedua sensor berupa nilai orientasi sudut x,y dan z yang kemudian diproses *flight controller* dan dibandingkan dengan nilai sudut yang diatur dalam kalibrasi PID. *flight controller* memintahkan *ESC* untuk mengubah kecepatan putaran motor drone agar posisi drone tetap seimbang sesuai dengan nilai PID yang diinput.



Gambar 10. Sistem kendali *quadcopter*

menentukan pola gerakan dengan mengubah kecepatan dan gaya dorong yang dihasilkan oleh motor. Diagram blok dari sistem kendali PID *Quadcopter* ditunjukkan pada Gambar 11.



Gambar 11. Diagram blok sistem PID *Quadcopter*

Pada Gambar 11 ditunjukkan blok diagram sistem kendali PID *quadcopter* untuk gerak *pitch*, *roll*, dan *altitude*. Nilai *set point* merupakan *input* dari *remote control* yang nilainya diubah menjadi derajat kemiringan, nilai tersebut kemudian dibandingkan dengan nilai derajat kemiringan *actual quadcopter* dari pembacaan sensor *gyroscope*. Hasil perbandingan dari kedua nilai tersebut kemudian diproses oleh *controller* PID yang kemudian hasilnya diproses oleh blok dinamika motor. Blok dinamika motor berisi persamaan matematis untuk menghasilkan kecepatan putaran motor sesuai dengan perintah *controller* PID yang masuk.

IV. HASIL DAN PEMBAHASAN

A. Hasil Implementasi *Hardware*

Hasil akhir implementasi *quadcopter* untuk bagian kerangka dan elektronika ditunjukkan pada Gambar 12.



Gambar 12. Hasil implementasi pemasangan komponen dan kerangka

Hasil implementasi menghasilkan *quadcopter* yang memiliki massa sebesar 1,6 Kg dengan ukuran *frame* 550 mm dan menggunakan *propeller* yang memiliki ukuran panjang 25 cm.

B. Hasil Pengujian dan Analisis PID

1) Altitude Hold

Fungsi pergerakan *altitude hold* adalah agar *quadcopter* dapat terbang dan mempertahankan ketinggiannya sesuai dengan nilai *set point* yang diberikan. Pergerakan *altitude hold* hanya memerlukan nilai *Kp*, karena hanya mengoreksi *error* ketinggian terbang agar dapat mendekati nilai *set point*.

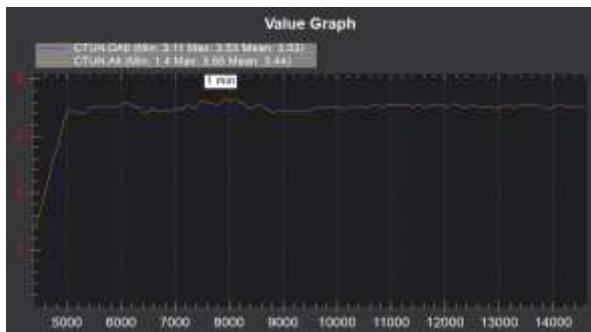
TABEL II. HASIL KESELURUHAN PERCOBAAN ALTITUDE HOLD

Percobaan ke	Nilai <i>Kp</i>	Rata –rata Error (meter)
1	0,5	0,085
2	1	0,042
3	1,5	0,020

Nilai *error* keseluruhan pada percobaan *altitude hold* dapat dilihat pada Tabel 2. Pada percobaan pertama ketinggian aktual *quadcopter* sudah dapat mencapai nilai *set point*, namun masih terjadi *overshoot* dan memiliki nilai rata -rata *error* 0,085 meter. Oleh karena ini ditambahkan nilai *Kp* pada percobaan kedua, dengan hasil *rise time* pada respon sistem semakin cepat mencapai *set point* yaitu selama 2,5 detik, akan tetapi sistem tidak dapat mempertahankan kestabilanya (*steady*) dan juga banyak menghasilkan nilai *error*.

Untuk memberikan efek koreksi pada *error* yang terjadi, nilai *Kp* terus ditambahkan hingga nilai rata-rata *error* mengecil, dilakukan percobaan ketiga pada Gambar 13 menunjukkan grafik dari percobaan ketiga, dapat dilihat nilai *rise time* yang cepat sebesar 0,50 detik dan sistem dapat melakukan efek koreksi yang cukup baik, hingga nilai rata-rata *error* nya 0,020 meter, nilai tersebut adalah respon sistem terbaik dari seluruh percobaan yang dilakukan.

Hasil akhir pengujian Kontrol PID untuk *altitude hold* dapat dilihat dari grafik *attitude* ketinggian yang berisi perbandingan nilai grafik *DAlt* (garis biru) sebagai nilai *set point*, dan *Alt* (garis hijau) sebagai nilai aktual. Nilai-nilai pada grafik tersebut memiliki satuan meter (m).



Gambar 13. Altitude hold nilai *Kp*=1,5

2) Pitch dan Roll

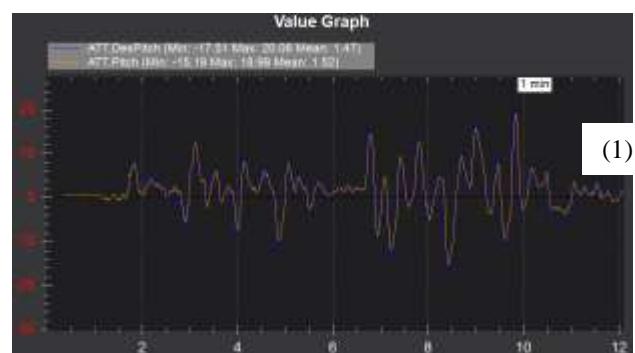
Untuk pergerakan *pitch* dan *roll* menggunakan nilai PID yang sama, nilai tersebut dimasukan secara bertahap dimulai dengan menggunakan control *P*, kemudian *PD* dan terakhir adalah *PID* hingga didapatkan kombinasi nilai yang tepat. Pengujian *PID* dilakukan dengan melihat grafik *attitude* yang dibaca oleh sensor *gyroscope* pada saat *quadcopter* sedang beroperasi.

TABEL III. HASIL KESELURUHAN PERCOBAAN PITCH DAN ROLL

Percobaan ke-	Nilai PID			Error (°)
	P	I	D	
1	0,25	0	0	1,56
2	0,25	0	0,04	1,60
3	0,25	0,15	0,04	0,10

Pada percobaan pertama diberikan nilai proposisional sebesar 0,25. Proposisional telah berhasil meredam sedikit nilai *overshoot* yang terjadi, namun tetap tidak menghilangkannya. Meskipun nilai *error* nya besar, namun nilai tersebut mampu meredam *overshoot* yang terjadi. Sesuai dengan karakteristiknya, kendali proposisional mampu meredam nilai *overshoot*, akan tetapi tidak dapat menghilangkannya. Untuk menghilangkannya diperlukan penambahan kendali derivatif.

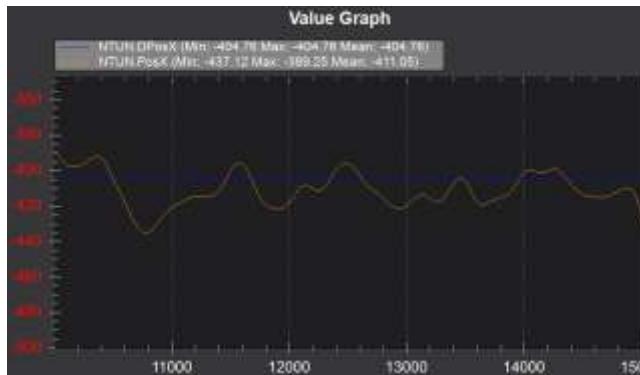
Hasil rata-rata *error* setelah ditambahkan nilai derivatif sebesar 0,04 dari percobaan ke 5 adalah 1,60°, nilai tersebut lebih besar dari sebelumnya. Untuk mengatasi masalah tersebut, maka ditambahkan kendali Integral pada percobaan berikutnya. Penambahan nilai integral sebesar 0,15 menghasilkan grafik pada Gambar 14. Hasil perpaduan nilai *PID* yang didapat dari hasil percobaan yang telah dilakukan adalah, nilai aktual dapat mengikuti nilai *set point*, memiliki respon yang cepat dan tidak terjadi *overshoot*.



Gambar 14. Grafik *pitch roll* nilai *P*=0,25, *D*=0,04 dan *I*=0,15

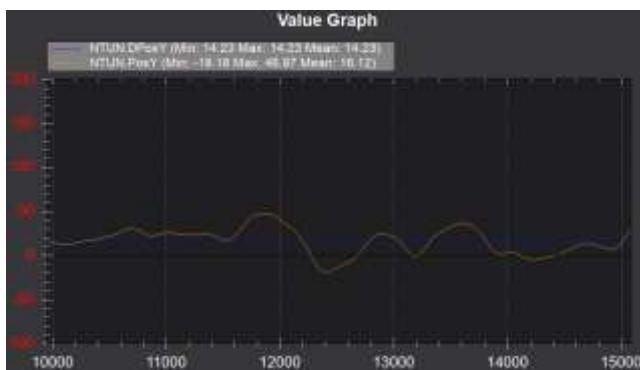
3) Position Hold

Pengujian sistem holding position dilakukan dengan melihat grafik nilai koordinat yang dibaca oleh GPS dan accelerometer. Grafik tersebut didapatkan melalui log data pada saat Drone skyscraper membawa arm robot yang sedang beroperasi. Gambar 15 adalah grafik untuk posisi x yang merupakan koordinat latitude, nilai positif untuk pergerakan kearah utara dan negatif kearah selatan.



Gambar 15. Grafik *latitude position hold*

Untuk koordinat *longitude* diperlihatkan grafik posisi Y pada Gambar 15, nilai positif untuk pergerakan kearah timur dan negatif kearah barat.



Gambar 16. Grafik *longitude position hold*

Berdasarkan gambar grafik dari pengujian *holding position*, maka didapatkan nilai *set point* dan nilai aktual dengan simpangan terjauh ditunjukkan pada Tabel 4. Dari grafik di atas didapatkan nilai-nilai *set point* beserta dengan nilai aktual posisi *quadcopter*, kedua nilai tersebut kemudian dibandingkan untuk mencari persentase *error* dengan menggunakan persamaan berikut

$$\%Error = \left[\frac{X_n - Y_n}{X_n} \right] \times 100\%$$

Keterangan :

$\%Error$ = Persentase kesalahan yang dikoreksi oleh sistem
 X_n = Nilai *Set Point*
 Y_n = Nilai Aktual/*Output*

Seluruh hasil $\%error$ dari data grafik posisi menghasilkan nilai rata-rata $\%error$ yang ditunjukkan pada Tabel 4

TABEL IV. NILAI RATA-RATA *ERROR POSITION HOLD*

Position	Rata-Rata %error
X (<i>Latitude</i>)	0,015
Y (<i>Longitude</i>)	0,55

V. PENUTUP

Cara kerja *quadcopter* adalah dengan memanfaatkan gerakan putaran motor dan baling-baling yang berlawanan arah dan menghasilkan putaran angin dibawah armada *quadcopter*, hal ini yang menjadikan *quadcopter* dapat terbang di udara. Seluruh putaran motor menyesuaikan kecepatan putarnya sesuai dengan perintah gerak yang diberikan. Perintah dari *remote control* diterima oleh *receiver*, yang kemudian diproses oleh *flight controller* dan dihasilkan *output PWM* kecepatan untuk keempat motor.

Sistem gerak dan kestabilan *quadcopter* dikendalikan menggunakan *controller PID*. *Drone skyscraper* dapat melakukan penerbangan yang stabil dengan kombinasi nilai PID sebagai berikut :

- Untuk gerak *altitude/hover*
Konstanta proposional : 1,5
- Untuk gerak *pitch* dan *roll*
Konstanta proposional : 0,25
Konstanta Integral : 0,15
Konstanta Derivatif : 0,04

Dengan pemberian nilai proposisional sebesar 1,5 pada pengendali *altitude*, didapatkan respon sistem yang memiliki *rise time* 0,50 detik dan sistem dapat mempertahankan kondisi stabilnya, dengan nilai rata-rata *steady state error* sebesar 0,20 meter. Dengan nilai kombinasi PID pada gerak *pitch* dan *roll*, didapatkan respon sistem yang nilai aktualnya dapat mengikuti nilai *set point* dengan respon yang cepat terhadap perubahan *set point* dan tidak terjadi *overshoot*.

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

Penentuan nilai *pitch*, *roll*, dan *altitude* lebih baik ditentukan dengan menggunakan simulasi MATLAB terlebih dahulu, agar mengurangi resiko *quadcopter* tidak terkontrol yang akan menyebabkan kerusakan pada alat. Pemilihan motor yang memiliki torsi lebih besar akan mampu mengangkat beban yang berat dengan konsekuensi *quadcopter* yang lambat, sedangkan motor bertorsi kecil akan mampu bergerak dalam kecepatan yang tinggi tetapi tidak untuk mengangkat beban yang besar. Penempatan posisi *arm robotic* disarankan mendekati titik *center of gravity*, agar motor tidak terlalu banyak bekerja untuk menyeimbangkan posisi dan kecepatan motor masih memiliki sisa torsi yang cukup untuk melakukan *take off* dan *hover*.

REFERENSI

- Setyawan, G., & Setiawan, E. (2015). Sistem Kendali Ketinggian Quadcopter Menggunakan PID. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)* , II, 125-131.
- Ilham, M. (2015). Rancang Bangun Pesawat UAV. Surabaya: Institut Teknologi Sepuluh Nopember.
- Rizky, M., Komarudin, M., & Trisanto, A. 2013. Sistem Kendali Holding Position Pada Quadcopter Berbasis Mikrokontroler Atmega 328p.
- Esa,A., Fahmizal, Nur,A.S.,Dhidik, P. 2019. Performance Evaluation of Balancing Bicopter using P, PI, and PID Controller. *Jurnal Teknik Eletktro*,Vol 11 no 2, 44-49.
- Ali, M. (2004). Pembelajaran Perancangan Sistem Kontrol PID dengan Software MATLAB. *Jurnal Edukasi Teknik Elektro* , I, 1-8.
- Hartono, B. (2018). Respon Gerak Kestabilan Dinamik Quadcopter Akibat Input Kendali pada Matra longitudinal dan Lateral. *Seminar Nasional Teknik Mesin*. Bandung: ITENAS.
- Habibi, G. A., Sarwoko, & Jati, A. N. (2013). Perancangan dan Analisis Otomasi Sistem Kendali Quadcopter. Bandung: Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom.
- quadkopters.com. (2018). *Sunny Sky 980KV Product*. Retrieved Agustus 18, 2020, from Quadkopters: <https://www.quadkopters.com/product/brushless-motors/sunnysky-2212-980kv-v2/>
- Lubis, J. A. (2017). Rancang Bangun Model Quadcopter Dengan Control PID . *JITEKH* , VI, 77-82.
- Hartman, D., Landis, K., Mehrer, M., Moreno, S. And Kim, J., 2014, Quadcopter Dynamic Modelling and Simulation (Quad-Sim) V. 1.00. <https://github.com/dch33/Quad-Sim>.