

PERANCANGAN SISTEM KONTROL NAVIGASI BEARING PADA QUADCOPTER DENGAN METODE PID (PROPORTIONAL, INTEGRAL, DERIVATIVE) SELF TUNING PSO (PARTICLE SWARM OPTIMIZATION)

Muhammad Surya Sulila^{*)}, Sumardi, and Munawar Agus Riyadi

Departemen Teknik Elektro, Universitas Diponegoro, Semarang
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)E-mail: msuryasulila@gmail.com}

Abstrak

Unmanned Aerial Vehicle (UAV) adalah pesawat tanpa awak yang dapat dikendalikan secara manual ataupun otomatis dari jarak jauh. Sistem navigasi UAV *quadcopter* salah satunya adalah membuat sistem kontrol *quadcopter* agar dapat stabil menghadap ke arah koordinat yang dituju dengan mengatur sudut putar sumbu vertikal (*yaw*) atau disebut navigasi *bearing* sehingga pada Penelitian ini dirancang sistem kontrol *Proportional Integral Derivative self tuning Particle Swarm Optimization*. Perancangan sistem navigasi *bearing* digunakan input berupa *Global Position System* untuk mengetahui koordinat *quadcopter*, sedangkan sensor kompas HMC5883L digunakan untuk mengetahui kondisi aktual sudut arah hadap *quadcopter*. Berdasarkan hasil pengujian respon sistem *quadcopter*, untuk dapat mengarah ke koordinat yang dituju dengan koordinat *quadcopter* tetap, *settling time* dicapai pada detik ke 6,4 dan *error* setelah *settling time* sebesar 5,4°. Berdasarkan pengujian dengan perubahan koordinat, didapatkan *error* rata-rata sebesar 7,9°. Berdasarkan pengujian dengan diberi gangguan didapatkan *error offset* rata-rata sebesar 1,89° dan mencapai *settling time* pada detik ke 4,1. Batasan nilai *self tuning* PSO yang terbaik didapat pada nilai $K_p = 0,15$ sampai 0,3, $K_i = 0,06$ sampai 0,6, dan $K_d = 0,005$ sampai $K_d = 0,1$. Nilai koefisien PSO yang digunakan adalah $C_1 = 1,5$, $C_2 = 2$ dan bobot inersia dari 0,7 sampai 1,2.

Kata kunci: Quadcopter, Navigasi Bearing, PID self tuning PSO.

Abstract

Unmanned Aerial Vehicle (UAV) is an unmanned aircraft can be controlled manually or automatically over long distance. UAV navigation of quadcopter system one of them is make the quadcopter control system to be able to stabilize facing towards destination coordinates by adjusting vertical axis rotation angle (*yaw*) or called bearing navigation so in this Research designed control system of Proportional Integral Derivative Self Tuning Particle Swarm Optimization. Design of bearing navigation system is used Global Position System input to determine coordinates of quadcopter, while HMC5883L compass sensor is used to determine actual condition of angle direction of quadcopter. Based on quadcopter response test results, to be able to lead coordinates of destination with fixed quadcopter coordinates, settling time is reached at 6.4 seconds and error after settling time of 5.4°. Coordinate change, average error is 7.9°. With disturbance average offset error of 1.89 ° and reach settling time in seconds to 4.1. The best PSO self tuning limits are obtained at $K_p = 0.15$ to 0.3, $K_i = 0.06$ to 0.6, and $K_d = 0.005$ to $K_d = 0.1$. Coefficient PSO used is $C_1 = 1.5$, $C_2 = 2$ and weight of inertia from 0.7 to 1.2.

Keywords: Quadcopter, Navigasi Bearing, PID self tuning PSO

1. Pendahuluan

Unmanned Aerial Vehicle (UAV) adalah tipe pesawat terbang tanpa awak yang dikendalikan secara manual ataupun otomatis dari jarak jauh. UAV saat ini dalam masa pengembangan yang pesat karena banyak digunakan untuk keperluan sipil dan militer. *Quadcopter* adalah salah satu jenis UAV yang komponen gerakannya berupa baling-baling yang berputar (*rotary wing*). *Quadcopter* merupakan

pengembangan dari *helicopter* yang hanya memiliki satu buah rotor. *Quadcopter* menggunakan empat buah rotor dengan kerangka yang menyilang. Mekanisme gerak *quadcopter* untuk maju, mundur, kanan dan kiri dengan cara mengurangi kecepatan dua buah motor sedangkan untuk berputar pada sumbu vertikal hanya mengurangi kecepatan satu buah motor [1].

Pengembangan sistem navigasi *quadcopter* salah satunya adalah membuat sistem kontrol *quadcopter* agar dapat

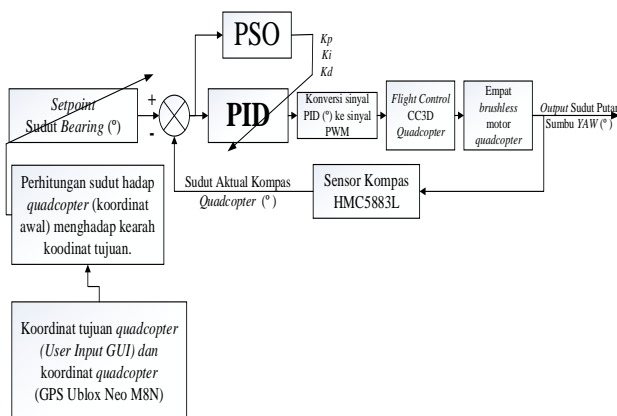
selalu menghadap koordinat lokasi yang dituju sehingga *quadcopter* jika diterbangkan kemudian bergerak maju ke depan akan sampai pada lokasi yang dituju. Parameter yang harus dikendalikan agar arah UAV stabil menghadap koordinat lokasi yang dituju adalah dengan mengatur sudut putar sumbu vertikal (*yaw*) *quadcopter* menghadap tujuan atau disebut sebagai *bearing*.

Dari ulasan Andrew dan Samuel sistem kontrol untuk *output* enam derajat kebebasan (maju atau mundur, naik atau turun, kiri atau kanan, *yaw*, *pitch*, *roll*) *quadcopter* terbaik salah satunya algoritma genetik dibandingkan kontrol lainnya seperti PID, LQR, LQG, *Backstepping*, Fuzzy Logic, *Neural Network* dan lain sebagainya [2]. Sistem kontrol yang dibandingkan berupa karakteristik adaptif, kepresisian, kecepatan respon konvergensi, kemudahan algoritma, *noise signal* dan lain sebagainya. *Particle Swarm Optimization* (PSO) memiliki metode optimasi kontrol dengan pencarian solusi secara populasi heuristik yang mirip seperti algoritma genetik [3]. Oleh karena itu dirancang sistem kontrol PID dengan *self tuning* PSO untuk dapat mengontrol navigasi *bearing quadcopter*.

2. Metode

2.1. Kontrol PID self tuning PSO

Perancangan pengendalian PID dengan *self tuning* PSO digunakan untuk dapat mengontrol sinyal PWM *yaw quadcopter* dengan adanya perubahan terus menerus *setpoint* dikarenakan adanya perubahan koordinat *quadcopter*. Penentuan nilai *self tuning* parameter yang digunakan pada PID adalah nilai parameter Kp, Ki dan Kd. Diagram blok proses kontrol PID *self tuning* PSO dapat dilihat pada Gambar 1.



Gambar 1. Diagram blok controller

Persamaan untuk *self tuning* PSO dapat dilihat berikut ini [4], [5], [6]:

- Persamaan perhitungan kecepatan partikel :
$$V_j(i) = \theta V_j(i-1) + c_1 r_1 [P_{best,j} - x_j(i-1)] + c_2 r_2 [G_{best} - x_j(i-1)]$$

$$j = 1, 2, \dots, N$$
 (1)

- Persamaan perhitungan bobot inersia :
$$\theta = \theta_{max} - \left(\frac{\theta_{max} - \theta_{min}}{error_{max}} \right) error_{actual}$$
 (2)

- Persamaan perhitungan posisi partikel :
$$X_j = X_j(i-1) + V_j(i)$$

$$j = 1, 2, \dots, N$$
 (3)

- Persamaan perhitungan konstanta Kp, Ki dan Kd :
$$Kp_j(i) = X_j = X_j(i-1) + V_j(i)$$

$$Ki_j(i) = X_j = X_j(i-1) + V_j(i)$$

$$Kd_j(i) = X_j = X_j(i-1) + V_j(i)$$

$$j = 1, 2, \dots, N$$
 (4)

Untuk dapat menyesuaikan dengan kontrol PID maka nilai bobot inersia menggunakan persamaan 2 sehingga nilai θ akan semakin kecil jika mendekati *setpoint* yang diberikan karena *error* akan mengecil jika mendekati *setpoint*. Nilai $P_{best,j}$ didapatkan dengan cara melakukan uji coba pada PID secara langsung sehingga ada perbandingan nilai diantara $X_j(i)$ dan $X_j(i-1)$ kemudian dicari nilai *error* terkecil dari keduanya. Sedangkan untuk mencari nilai G_{best} maka nilai dari seluruh individu $X_j(i)$ dalam satu iterasi di uji coba langsung pada kontrol PID kemudian setiap kali uji coba dibaca nilai *error* yang dihasilkan kemudian dibandingkan mana nilai $X_j(i)$ terbaik yang memiliki nilai *error* terkecil maka akan menjadi nilai G_{best} .

2.2. Modul GPS U-Blox Neo M8N

Perancangan GPS U-Blox Neo M8N membutuhkan tegangan kerja 5 Volt. Menggunakan sistem komunikasi serial dengan *baudrate* 9600 dengan data yang dikirim berupa data NMEA GGA yang diurai dan didapat berupa data *degrees minutes* (DM) kemudian dirubah menjadi *degree decimal* (DD) untuk mendapatkan nilai *longitude* dan *latitude* secara DD [7]. Format pada GPS dapat dilihat pada contoh berikut:

```
$GPGGA,*****,ddmm.mmmmm,Direction,dddmm.mm  
mmm,Direction,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*  
  
$GPGGA,*****,0703.76714,S,11026.84100,E,*,*,*,*,*  
*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
```

Sehingga data menjadi:

- Latitude format DM : 0703.76714 sign “S”
- Latitude format DD : -7.062785
- Longitude format DM : 11026.84100
- Latitude format DD : 110.447350

2.3. Modul Sensor Magnetometer HMC5883L

Sensor kompas HMC5883L menggunakan sistem komunikasi data berupa I2C yang memiliki dua *port* data yaitu SDA dan SCL. Data yang didapat dari pada kompas

yaitu 3 *axis* data. Data yang didapat memiliki panjang 16 bit data. Pengolahan data dari *axis* kompas yaitu dengan melakukan perhitungan arcus tan pada data *xadjusted* dan *yadjusted*. Data *xadjusted* pada alamat 0x3c register 0x03 sampai 0x04 sedangkan pada *yadjusted* pada 0x07 sampai 0x08. Setelah didapat nilai radian arcus tan maka dirubah menjadi sudut dengan dikalikan 180/pi kemudian ketika sudut lebih dari nol maka data kompas menjadi 360 dikurangi data kompas dan ketika sudut kurang dari nol maka data kompas ditambahkan 360 [8].

2.4. Filter Regresi Polinomial

Filter regresi polinomial digunakan untuk memperbaiki nilai kompas HMC5883L dengan nilai yang seharusnya karena pada sensor kompas HMC5883L dapat terjadi *error* yang besar yaitu mencapai 22° antara pembacaan sensor kompas dengan nilai yang seharusnya. Sehingga perlunya ada perbaikan nilai menggunakan regresi polinomial orde empat dan orde dua menggunakan matlab, dengan masukan berupa “x” dari sensor kompas dan keluaran berupa nilai kompas yang seharusnya “y” maka akan mendapat persamaan polinomial orde 4. Berikut ini persamaan mencari fungsi orde 2 [9]:

- Fungsi pendekatan :

$$y = a_0 + a_1x + a_1x^2 + .. + a_nx^n (x + a)^n$$

(5)

y = nilai data yang seharusnya, x = data pengukuran, a_n = kostanta fungsi regresi polinomial, n adalah banyaknya data.

- Persamaan Orde 2 :

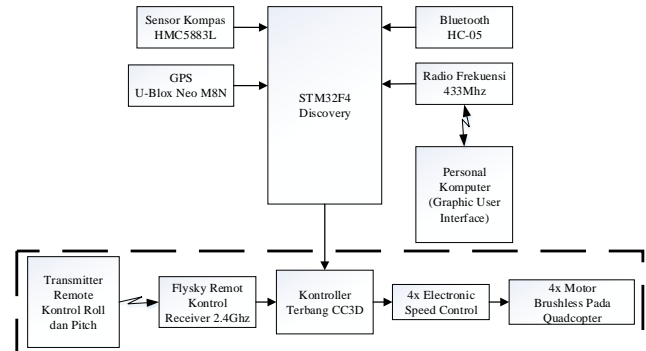
$$\begin{aligned} na_0 + (\sum x_i) a_1 + (\sum x_i^2) a_2 &= \sum y_i \\ (\sum x_i) a_0 + (\sum x_i^2) a_1 + (\sum x_i^3) a_2 &= \sum x_i y_i \\ (\sum x_i^2) a_0 + (\sum x_i^3) a_1 + (\sum x_i^4) a_2 &= \sum x_i^2 y_i \end{aligned}$$

(6)

2.5. Perancangan Perangkat Keras

Perancangan perangkat keras utama terdapat pada mikrokontroler ARM STM32F4 Discovery yang mengelola *input* maupun *output* sistem kontrol. *Input* pada mikrokontroler yaitu berupa sensor kompas HMC5833L dan GPS Neo M8N. Sensor kompas digunakan untuk mengetahui posisi sudut hadap *quadcopter* dari sumbu utara bumi dan untuk GPS digunakan untuk mengetahui koordinat dari pada *quadcopter* kedua *input* tersebut digunakan untuk proses navigasi *bearing*. Navigasi *bearing* diperlukan adanya posisi awal dan posisi tujuan serta perlunya posisi sudut hadap *quadcopter* agar mengetahui bahwa *quadcopter* selalu menghadap ke arah tujuan. *Output quadcopter* untuk proses kontrol yaitu PWM untuk sinyal kontrol pada CC3D sedangkan pada *bluetooth* untuk melihat data hasil pembacaan sensor, *gps*,

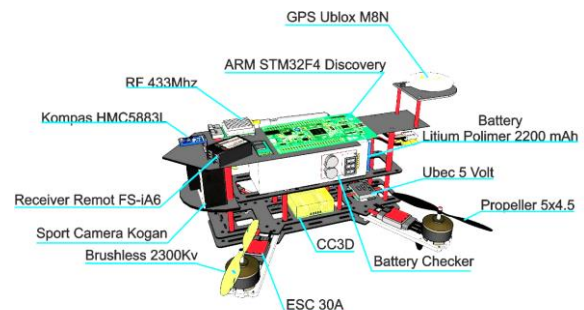
sistem kontrol dan lain sebagainya sedangkan untuk radio frekuensi digunakan untuk memonitor hasil sistem kontrol dari pada *quadcopter* dan merubah koordinat tujuan *quadcopter* pada penerima radio frekuensi berupa GUI pada personal komputer. Sedangkan untuk remot kontrol pada CC3D digunakan secara manual untuk mengontrol *quadcopter* secara *pitch* dan *roll*. Keseluruhan perancangan dapat dilihat pada Gambar 2 Kotak garis putus-putus tidak termasuk kedalam pembahasan perancangan sistem kontrol dari penelitian ini.



Gambar 2. Diagram perancangan sistem perangkat keras.

2.6. Desain Mekanik dan Peletakan Komponen Elektronika

Sistem mekanik pada *quadcopter* menggunakan 4 buah lengan yang digunakan untuk brushless motor. Desain mekanik dibuat menggunakan karbon fiber dengan tipe *quadcopter racing ZMR 250* yang banyak beredar di pasaran dengan tambahan papan *frame* untuk peletakan komponen elektronika. Papan *frame* untuk komponen elektronika menggunakan bahan fiber *Printed Circuit Board (PCB)* dengan menggunakan papan *frame* sekaligus sebagai PCB untuk peletakan komponen elektronika membuat desain mekanik *quadcopter* menjadi lebih sederhana. Desain mekanik dan peletakan komponen elektronika pada *quadcopter* pada Gambar 3.



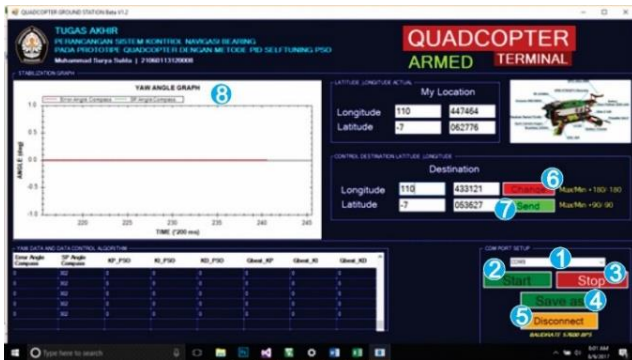
Gambar 3. Desain mekanik dan posisi komponen elektronika.

2.7. Perancangan Perangkat Lunak

Perancangan perangkat lunak terdiri dari perancangan perangkat lunak untuk mikrokontroler dan *graphic user*

interface. Perangkat lunak untuk mikrokontroler ARM STM32F4 Discovery menggunakan aplikasi Keil Uvision untuk proses programming seperti sistem kontrol PID *self tuning* PSO, tapis filter regresi polinomial untuk sensor kompas, parsing data dan perubahan satuan koordinat GPS dari degree minute ke degree decimal, algoritma pencarian nilai *error* sudut *bearing* tekecil, pengaturan PWM, *input output bluetooth* dan radio frekuensi.

Perancangan perangkat lunak untuk graphic user interface menggunakan Visual studio C# dengan fitur dapat memonitor parameter sistem kontrol dan penggantian koordinat arah tujuan *quadcopter* melalui radio frekuensi. Perancangan Visual Studio C# mengambil data melalui komunikasi serial dengan *baudrate* 57600 setelah pengambilan data kemudian data diurai masing-masing menjadi data tunggal. Parsing data berupa data *error* sudut *bearing*, *setpoint bearing*, parameter Kp, Ki dan Kd, nilai PSO Gbest Kp, Gbest Ki dan Gbest Kd, koordinat lokasi *quadcopter* latitude dan longitude serta kontrol mengganti koordinat arah tujuan *quadcopter*. GUI dapat dilihat pada Gambar 4.



Gambar 4. Tampilan perancangan GUI.

2.8. Perancangan Arah Putar Sudut Yaw dengan Error terkecil

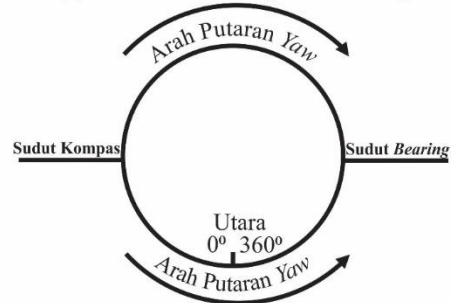
Perancangan arah putar yang tepat untuk sudut *yaw* dengan *error* terkecil digunakan untuk menentukan *quadcopter* berputar secara searah jarum jam atau *quadcopter* berputar berlawanan jarum jam. Perhitungan untuk menentukan putaran *quadcopter* dengan cara menghitung nilai sudut selisih antara nilai aktual *quadcopter* dengan nilai *setpoint* yang diberikan kemudian memilih *error* yang lebih kecil searah jarum jam atau berlawanan jarum jam untuk menuju sudut *setpoint* yang diberikan.

Langkah-langkah untuk menentukan *error* terkecil Diketahui:

- Error* positif digunakan untuk searah jarum jam.
- Error* negatif digunakan untuk berlawanan jarum jam.
- Error_atas* adalah *error* yang tidak melewati sudut 0° atau 360°.

- Error_360* adalah *error* yang melewati sudut 0° atau 360.

$$\text{Error_atas} = \text{Sudut Bearing} - \text{Sudut Kompas}$$



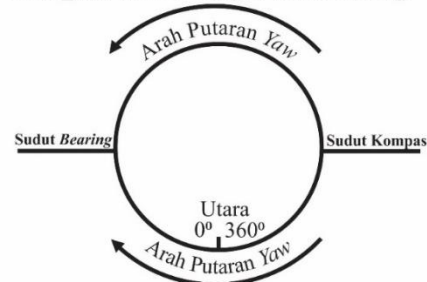
$$\text{Error_360} = (360 - \text{Sudut Bearing}) + \text{Sudut Kompas}$$

Gambar 5. Nilai sudut *bearing* lebih kecil dari nilai sudut kompas.

Algoritma untuk fungsi pertama :

- Jika nilai *bearing* lebih besar dari nilai kompas seperti pada Gambar 5 maka menghitung nilai $\text{Error_atas} = \text{Bearing} - \text{Data_Kompas}$ dan $\text{Error_360} = (360 - \text{Bearing}) + \text{Data_kompas}$
- Jika *Error_atas* lebih kecil dari *Error_360* maka $\text{error_PID} = \text{Error_atas}$ dengan arah putar searah jarum jam.
- Jika *Error_atas* lebih besar dari *Error_360* maka $\text{error_PID} = \text{Error_360} * -1$ dengan arah putar berlawanan jarum jam.

$$\text{Error_atas} = \text{Sudut Kompas} - \text{Sudut Bearing}$$



$$\text{Error_360} = (360 - \text{Sudut Kompas}) + \text{Sudut Bearing}$$

Gambar 6. Nilai sudut *bearing* lebih kecil dari nilai sudut kompas.

Algoritma untuk fungsi kedua :

- Jika nilai *bearing* lebih kecil dari nilai kompas seperti pada Gambar 6 maka menghitung nilai $\text{Error_atas} = \text{Data_Kompas} - \text{Bearing}$ dan $\text{Error_360} = (360 - \text{Data_kompas}) + \text{Bearing}$.
- Jika *Error_atas* lebih kecil dari *Error_360* maka $\text{error_PID} = \text{Error_atas} * -1$ dengan arah putar berlawanan jarum jam.
- Jika *Error_atas* lebih besar dari *Error_360* maka $\text{error_PID} = \text{Error_360}$ dengan arah putar searah jarum jam.

3. Hasil dan Analisa

3.1. Pengujian GPS U-Blox Neo M8N

Pengujian data GPS dilakukan untuk mengetahui akurasi dari pada GPS tersebut karena semakin dekat jarak antara *quadcopter* dengan tujuan akan membuat *quadcopter* tidak mengetahui bawah titik tujuan sudah di dekatnya ataupun jauh dari dirinya sendiri. Berikut tabel hasil pengujian GPS pada Tabel 1.

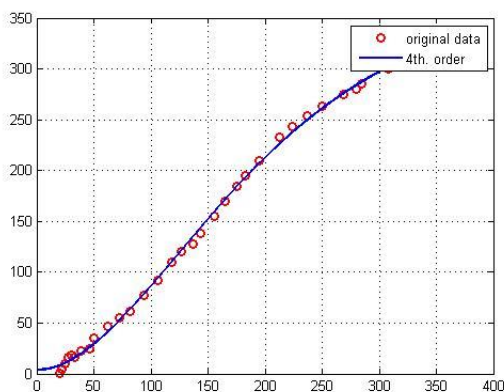
Tabel 1. Kesimpulan pengujian GPS

Jumlah Data GPS	Maksimal Error (Meter)	Minimal Error (Meter)	Rata-Rata Error (Meter)
2850	6,57	1,33	3,86

Hasil pengujian GPS didapatkan nilai *error* maksimal mencapai 6,57 meter, *error* minimal 1,33 meter dan rata-rata *error* 3,86 meter. Nilai *error* maksimal 6,57 meter menyatakan jika *quadcopter* berada pada radius dengan diameter 6,57 meter pada titik tujuan maka *quadcopter* tidak akan terjadi proses kontrol karena berubah nilai posisi *quadcopter* tidak diketahui perubahannya. Oleh karena itu *quadcopter* dapat efektif bekerja pada radius yang lebih besar dari diameter *error* maksimal 6,57 meter jarak *quadcopter* dari titik tujuan.

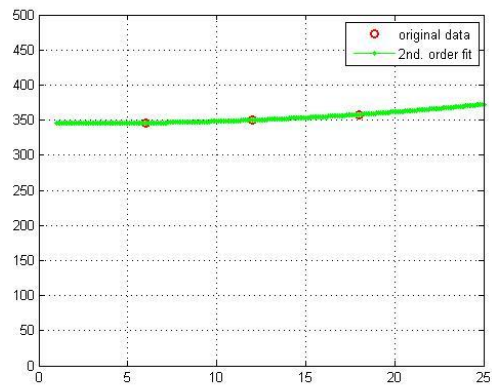
3.2. Pengujian Sensor Kompas HMC5883L

Pengujian validitas sensor kompas HMC5883L dilakukan menggunakan pembandingan kompas digital *smartphone*. Didapat data hasil percobaan sensor kompas HMC5883L dengan kompas digital *smartphone* diperbaiki dengan regresi polinomial orde empat pada Gambar 7 dan orde dua pada Gambar 8.



Gambar 7. Grafik regresi polinomial orde 4 untuk 20° sampai 355°.

Sedangkan untuk pengujian sensor kompas untuk nilai lebih dari 0° sampai kurang dari 19° grafik regresi polinomial orde 2 pada Gambar 8.



Gambar 8. Grafik regresi polinomial orde 2 untuk 0° sampai 19°.

Dengan menggunakan regresi polinomial orde 4 dan orde 2 untuk memperbaiki nilai sensor kompas HMC5883L maka didapat persamaannya yaitu:

Untuk $x_{20^{\circ}-355^{\circ}}$:

$$y = 0.00000005242854118x^4 - 0.00004678548355x^3 + 0.01275813135x^2 + 0.05976890286 + 5.921211414 \quad (7)$$

Untuk $x_{0^{\circ}-19^{\circ}}$:

$$y = 0,0556x^2 - 0,3333x - 346 \quad (8)$$

$y = Output$ regresi polinomial orde 2
 $x = Input$ sensor kompas HMC5883L

Dengan persamaan orde 4 dan orde 2 diatas maka nilai kompas dapat diperbaiki nilainya sehingga mendekati nilai sebenarnya. Berikut hasil pengujian setelah nilai sensor kompas diperbaiki dengan persamaan orde 4 dan orde 2 pada Tabel 2.

Tabel 2. Pengujian nilai kompas hasil perbaikan regresi polinomial.

No	Sensor Kompas (° Utara)	Kompas Digital (° Utara)	Error setelah perbaikan (° Utara)
1	20	24	2
2	345	348	3
3	359	358	1
4	10	5	5
5	239	245	6
Rata-rata			3,4

Dapat dilihat pada Tabel 2 *error* maksimal yang terjadi antara sensor kompas HMC5883L dengan sensor kompas digital setelah melalui proses perbaikan menggunakan regresi polinomial orde empat dan orde dua memiliki *error* terbesar 6° yang sebelumnya *error* terbesar mencapai 22° dan rata-rata *error* hanya 3,4°.

3.3. Pengujian Arah Yaw Terkecil

Pengujian ini digunakan untuk menentukan arah putar sudut sumbu yaw dengan error terpendek pada sistem. Hasil pengujian dilihat pada Tabel 3.

Tabel 3. Pengujian arah yaw error terkecil

No	Sudut Aktual (°)	SP (°)	Error CW (°)	Error CCW (°)	Arah Gerak	Hasil
1	77	285	208	152	CCW	Sesuai
2	275	121	206	154	CCW	Sesuai
3	126	185	59	301	CW	Sesuai
4	82	267	185	175	CCW	Sesuai

Keterangan :

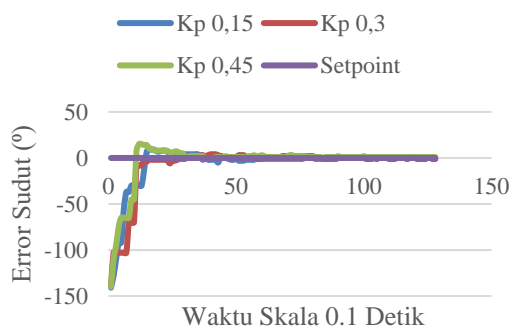
- Berlawanan jarum jam "Counterclockwise (CCW)".
- Searah jarum jam "Clockwise (CW)".
- SP (setpoint).

Berdasarkan hasil dari Tabel 3. sistem kontrol yang dibuat dapat menentukan arah putar sumbu yaw yang memiliki error terkecil dari setpoint. Sehingga jika error dengan peputaran searah jarum jam lebih besar dibandingkan peputaran berlawanan jarum jam maka sistem akan memilih berputar berlawanan jarum jam karena memiliki nilai error yang lebih kecil maupun sebaliknya jika error berputarnya sistem searah jarum jam lebih kecil dibandingkan error berputarnya sistem berlawanan jarum jam maka sistem akan memilih untuk berputar secara searah jarum jam.

3.4. Pengujian Parameter Kontrol PID PSO

3.4.1. Pengujian Menentukan Batas Nilai Kp

Pengujian variasi menentukan batasan nilai Kp self tuning PSO menggunakan nilai 0.01 sampai 0.15 kemudian 0.15 sampai 0.3 dan 0.3 sampai 0.45. Respon sistem dari ketiga percobaan dapat dilihat pada Gambar 9.



Gambar 9. Respon sistem perbandingan nilai Kp self tuning PSO.

Respon sistem untuk nilai Kp 0,01 sampai 0,15 memiliki karakteristik sering terjadinya osilasi. Respon sistem untuk nilai Kp 0,3 sampai 0,15 memiliki karakteristik lebih stabil

dibandingkan nilai kedua lainnya karena memiliki rata-rata error offset yaitu 1,1° dan settling time pada detik ke 2,6, terlihat tidak ada overshoot yang besar dibandingkan kedua lainnya. Pada respon sistem untuk nilai Kp 0,3 sampai 0,45 terlihat bahwa overshoot sangat besar terjadi dibandingkan nilai kedua lainnya mencapai nilai error 8°.

3.4.2. Pengujian Menentukan Batas Nilai Ki dan Kd

Pengujian menentukan batasan nilai Ki dan Kd menggunakan nilai konstan Kp terbaik dari percobaan yang didapat sebelumnya yaitu nilai terbaik untuk Kp adalah 0.15 sampai 0.3 sedangkan untuk menentukan batasan nilai Ki dan Kd dilakukan sebanyak sembilan kali percobaan dengan variasi nilai sebagai berikut:

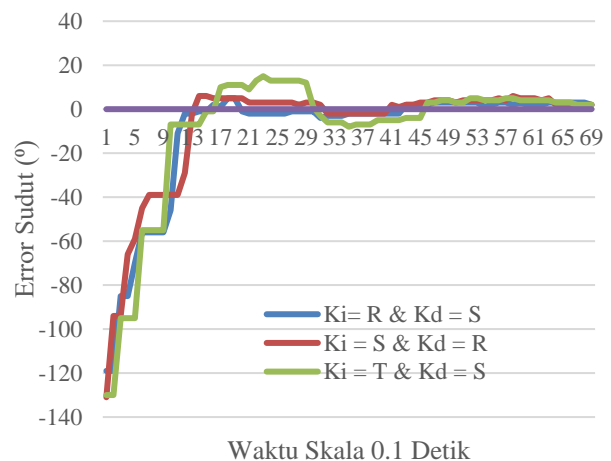
Percobaan nilai untuk Ki:

- Nilai rendah 0,06 sampai 0,6 kelompok rendah simbol (R).
- Nilai sedang 0,6 sampai 1,2 kelompok sedang simbol (S).
- Nilai tinggi 1,2 sampai 1,8 kelompok tinggi simbol (T).

Percobaan nilai untuk Kd:

- Nilai rendah 0,0005 sampai 0,005 kelompok rendah simbol (R).
- Nilai sedang 0,005 sampai 0,01 kelompok sedang simbol (S).
- Nilai tinggi 0,01 sampai 0,015 kelompok tinggi simbol (T).

Kesimpulan perbandingan nilai terbaik dari hasil pengujian sembilan kombinasi nilai rendah sampai nilai tinggi dapat dilihat pada Gambar 10.



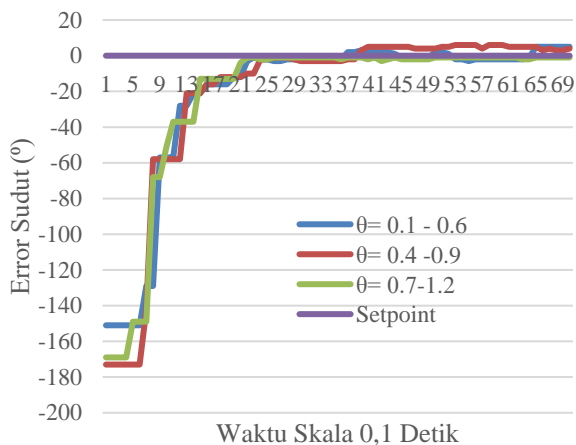
Gambar 10. Perbandingan tiga nilai Ki dan Kd terbaik dari sembilan kombinasi

Respon sistem untuk nilai Ki rendah dan Kd sedang memiliki karakteristik yang stabil dibandingkan lainnya karena rise time pada detik ke 1,1, rata-rata offset 2,31° dan

settling time pada detik ke 1,2. Respon sistem untuk nilai K_i sedang dan K_d rendah memiliki karakteristik *rise time* pada detik ke 1,3, rata-rata *error offset* $3,28^\circ$, dan *settling time* pada detik ke 1,3. Respon sistem untuk nilai K_i tinggi dan K_d sedang memiliki *rise time* pada detik ke 1, nilai *error offset* $3,83^\circ$, dan *settling time* pada detik ke 3,9°. Maka didapat nilai terbaik dari batas maksimal dan minimal K_i dan K_d adalah untuk K_i dari 0,06 sampai 0,6 sedangkan untuk K_d dari 0,005 sampai 0,1.

3.5. Pengujian bobot inersia

Menentukan nilai koefisien dari pada inersia θ digunakan untuk penentuan besaran nilai update dari pada PSO. Penentuan koefisien digunakan rentang nilai dari nilai rendah yaitu 0,1 sampai 0,6 kemudian 0,4 sampai 0,7 dan nilai tinggi 0,7 sampai 1,2. Hasil pengujian respon sistem terhadap perubahan nilai inersia dapat dilihat pada Gambar 11.



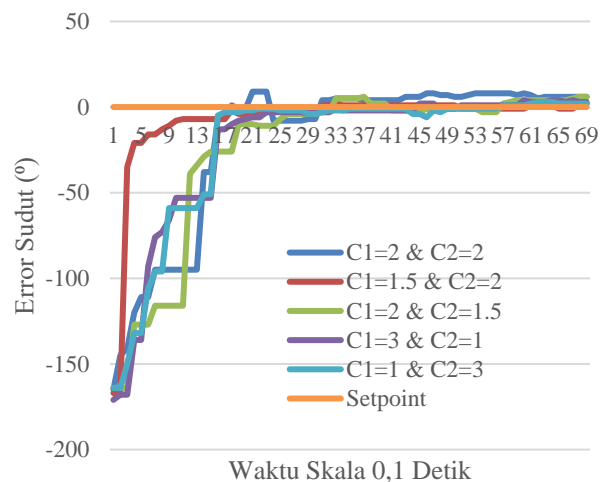
Gambar 11. Respon sistem terhadap variasi nilai inersia

Pada Gambar 11 hasil pengujian untuk nilai inersia dari 0,1 sampai 0,6 memiliki karakteristik respon sistem sering terjadi osilasi dan adanya *steady state* pada akhir respon sistem. Hasil pengujian untuk nilai inersia dari 0,4 sampai 0,9 didapat karakteristik respon sistem *steady state* karena pada awal *rise time* saja respon sistem dapat mencapai nilai *setpoint*. Hasil pengujian untuk nilai inersia dari 0,7 sampai 1,2 memiliki karakteristik respon sistem yang memiliki *rise time* pada detik ke 1,5, *error* rata-rata *offset* $1,34^\circ$, dan mencapai *settling time* pada detik ke 2,1.

3.6. Pengujian Nilai C1 dan C2

Pengujian penentuan nilai C_1 dan C_2 digunakan untuk menentukan nilai terbaik dari pada pengaruh kognitif koefisien C_1 terhadap nilai kecepatan update karena pada C_1 memiliki pengaruh terhadap nilai perubahan dari setiap partikel itu sendiri sedangkan untuk nilai C_2 digunakan untuk menentukan nilai terbaik dari pengaruh sosial

koefisien C_2 terhadap kecepatan update karena pada C_2 memiliki pengaruh terhadap perubahan nilai dari seluruh partikel. Pengujian C_1 dan C_2 dengan menggunakan variasi nilai $C_1=2$ dan $C_2=2$, $C_1=1,5$ dan $C_2=2$, $C_1=2$ dan $C_2=1,5$, $C_1=3$ dan $C_2=1$, $C_1=1$ dan $C_2=3$. Hasil pengujian respon sistem terhadap pengaruh C_1 dan C_2 dapat dilihat pada Gambar 12.

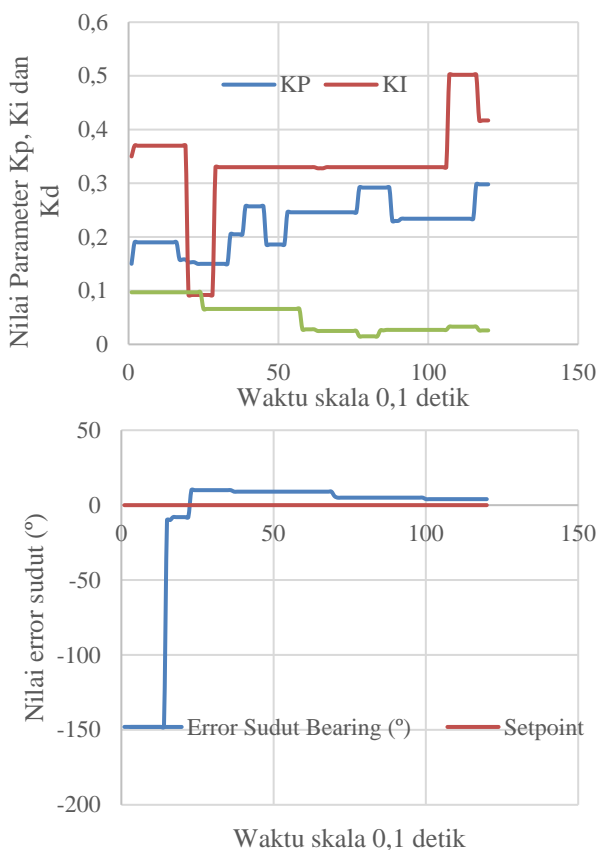


Gambar 12. Respon sistem variasi nilai C_1 dan C_2

Hasil respon sistem untuk C_1 dan C_2 bernilai 2 terjadi *overshoot* yang tinggi mencapai *error* 9° , dibandingkan dengan variasi nilai lainnya dan terjadi *steady state* pada akhir respon sistem. Respon sistem untuk $C_1=1,5$ dan $C_2=2$ memiliki karakteristik respon sistem dengan *rise time* pada detik ke 0,6, rata-rata *offset* $2,16^\circ$, dan *settling time* terjadi pada detik ke 1. Respon sistem dengan nilai $C_1=2$ dan $C_2=1,5$ memiliki karakteristik lambat saat proses *rise time* pada detik ke 1,9. Respon sistem pada $C_1=3$ dan $C_2=1$ memiliki karakteristik lambat mencapai *rise time* pada detik ke 1,6 dan *settling time* pada detik ke 1,9. Respon sistem untuk $C_1=1$ dan $C_2=3$ memiliki karakteristik lambat mencapai *rise time* pada detik ke 1,6.

3.7. Pengujian Perubahan Parameter PID Hasil Self tuning PSO

Pengujian ini dilakukan untuk mengetahui perubahan parameter K_p , K_i dan K_d dari hasil *self tuning* PSO. Pada pengujian ini parameter yang diperlukan untuk proses *self tuning* telah ditentukan sesuai percobaan terbaik dari pengujian sebelumnya. Penggunaan parameter terbaik didapat $K_p = 0,15$ sampai 0,3, $K_i = 0,06$ sampai 0,6 dan $K_d = 0,005$ sampai $K_d = 0,1$. Nilai koefisien terbaik untuk $C_1=1,5$ dan $C_2=2$ dan untuk pembobotan inersia digunakan dari 0,7 sampai 1,2. Respon sistem dengan perbandingan perubahan parameter pada Gambar 13.



Gambar 13. Respon sistem pengujian perubahan parameter pada PID self tuning PSO.

Hasil pengujian pada Gambar 13 menunjukkan nilai Kp bertambah sebelum mencapai *rise time* untuk mempercepat respon sistem. Nilai Kp berkurang setelah mencapai *rise time* untuk mengurangi adanya *overshoot*. Nilai Kp kembali bertambah setelah *steady state* digunakan untuk menghilangkan *error steady state*. Nilai Ki bertambah sebelum mencapai *rise time* untuk mempercepat respon sistem. Nilai Ki berkurang saat setelah mencapai *rise time* untuk mengurangi terjadinya *overshoot* pada sistem. Nilai Ki bertambah kembali setelah *steady state* digunakan untuk mengurangi *error steady state*. Nilai Kd bertambah pada saat *rise time* untuk meredam sistem kontrol P dan I sehingga tidak terjadi *overshoot*. Nilai Kd terus-menerus berkurang setelah *rise time* karena dapat menyebabkan terjadinya *steady state* terus-menerus jika Kd tetap ataupun bertambah.

3.8. Hasil Implementasi Keseluruhan Sistem

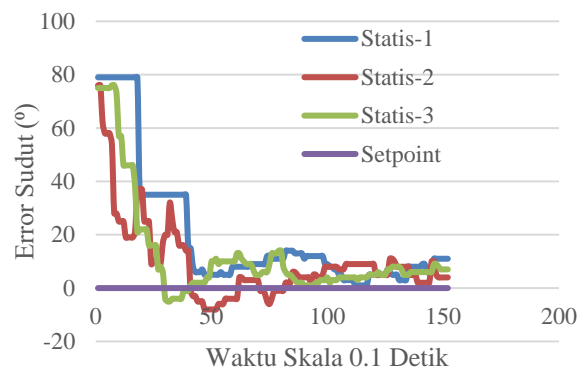
3.8.1. Pengujian Statis

Pengujian statis dilakukan di lapangan Woodball UNDIP. Pengujian dilakukan dengan membiarkan *quadcopter* untuk tetap menghadap arah tujuannya tanpa perlu diberi gangguan untuk bergerak secara *roll* atau horizontal. Jarak antara *quadcopter* dengan jarak yang dituju berkisar 30 meter karena memungkinkan untuk mendapat adanya

perubahan nilai dari pada pembacaan GPS karena dari validasi nilai GPS memiliki *error* maksimal pembacaan yaitu sebesar 6,54 Meter. Pada lokasi *quadcopter* yaitu *longitude* 110,432945 dan *latitude* -7,053425 sedangkan untuk tujuannya yaitu *longitude* 110,433121 dan *latitude* -7,053627. Pengujian statis dilakukan sebanyak tiga kali percobaan dengan titik awal *quadcopter* dan tujuan yang sama dapat dilihat pada Gambar 14 dan respon sistem kontrol dapat dilihat pada Gambar 15.



Gambar 14. Lokasi pengujian Lapangan Wooball UNDIP.



Gambar 15. Pengujian di lapangan secara statis.

Dari pengujian di lapangan secara statis didapatkan hasil *rise time* paling lambat pada detik ke 4,2 untuk pengujian statis-1 dan memiliki *error* terbesar setelah *rise time* mencapai 14° . Pada pengujian statis-2 terjadi osilasi saat mendekati *setpoint* sehingga *rise time* pada detik ke 4,1, *settling time* sesaat pada detik ke 6,4 sampai 7,3 dan rata-rata *error* setelah *settling time* sebesar $5,4^\circ$. Hasil *rise time* paling cepat pada detik ke 2,7 untuk pengujian statis-3. Pengujian statis-3 terjadi *settling time* sesaat pada detik ke 8,6 sampai 9,9, memiliki rata-rata *error* setelah *settling time* sesaat $4,71^\circ$. Percobaan terbaik didapat pada statis-2 walaupun terjadi osilasi setelah *rise time* disebabkan adanya noise dari pengujian di lapangan berupa angin yang arah dan besarnya tidak diketahui.

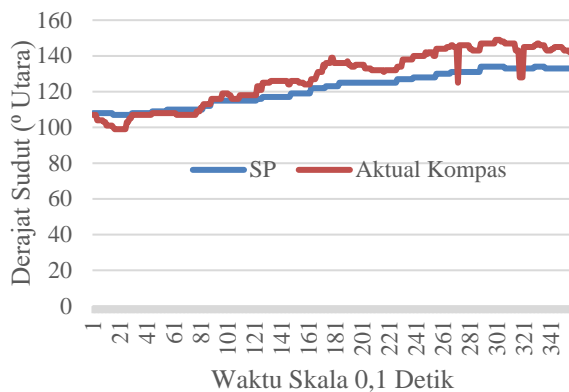
3.8.2. Pengujian Dinamis

Pengujian dinamis dilakukan di lapangan Woodball UNDIP. Pengujian dinamis dilakukan dengan mengerakan *quadcopter* secara horisontal atau *roll* sepanjang 21,51

meter. Titik-titik koordinat pengujian untuk awal bergerak *quadcopter* yaitu latitude -7,053505 dan longitude 110,43278 kemudian titik koordinat akhir pengujian *quadcopter* yaitu latitude -7,053315 dan longitude 110,432838. Titik tujuan dari *quadcopter* adalah menghadap koordinat latitude -7,053627 dan longitude 110,433121. Gambar keseluruhan titik koordinat dan jarak antara titik dapat dilihat pada Gambar 16 dan respon sistem kontrol dapat dilihat pada Gambar 17.



Gambar 16. Titik-titik pengujian dinamis.



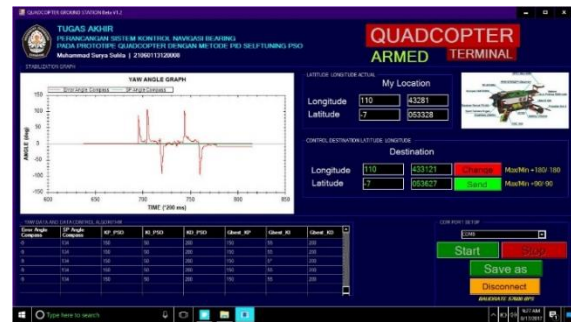
Gambar 17. Pengujian di lapangan bergerak secara dinamis

Pengujian dinamis di lapangan dapat dilihat bahwa *setpoint bearing* perlahan-lahan menaikkan nilainya karena pada percobaan ini *quadcopter* bergerak secara horisontal kearah kiri sehingga akan menambah nilai sudut dari pada *setpoint bearing*. Pada *quadcopter* secara aktual sistem kontrol dapat mengikut nilai *setpoint bearing* yang berubah-ubah dengan nilai *error* terbesar setelah mendekati *setpoint* adalah 16° , *error* rata-rata *offset* adalah $7,9^\circ$ dan *error* minimal yang terjadi adalah 1° . Pada sistem kontrol dengan adanya perubahan *setpoint* dapat dilihat bahwa *steady state* dapat terjadi pada derajat sudut yang lebih dari *setpoint* ataupun kurang dari pada *setpoint*. *Error* terbesar mencapai 16° dapat terjadi karena pengujian di lapangan terbuka sehingga adanya angin yang tidak diketahui kecepatan dan arahnya.

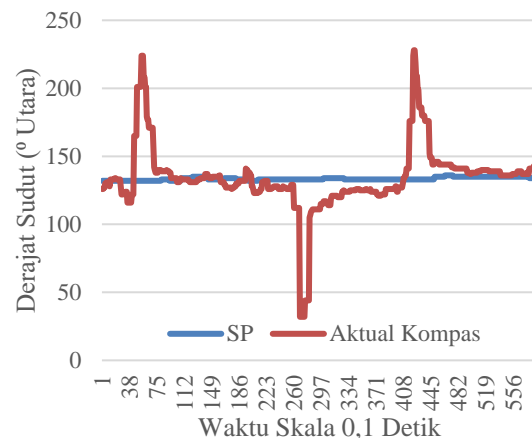
3.8.3. Pengujian diberi gangguan

Pengujian diberi gangguan dilakukan di lapangan Woodball UNDIP. Gangguan yang diberikan yaitu memutar secara langsung *quadcopter* kearah berlawanan

dari *setpoint* baik searah jarum jam maupun berlawanan jarum jam. Pengujian ini dilakukan untuk mengetahui bahwa sistem kontrol dapat mengatasi adanya *error* sudut yang cukup tinggi dari adanya gangguan. Pengujian *quadcopter* berada pada koordinat longitude 110,43281 dan latitude -7,053328 dengan *bearing* atau *setpoint* 134° dilihat pada Gambar 18 GUI pengujian diberi gangguan dan grafik respon sistem pada Gambar 19.



Gambar 18. GUI Pengujian diberi gangguan.



Gambar 19. Pengujian di lapangan dengan diberi gangguan.

Dari pengujian terhadap *quadcopter* yang diberi gangguan, didapatkan *rise time* untuk gangguan pertama mencapai detik ke 1,9, yang kedua pada detik ke 5,8, dan yang ketiga pada detik ke 3,5, *Settling time* untuk gangguan pertama mencapai detik ke 4,1, yang kedua terjadi *steady state* dan yang ketiga detik ke 7,3. Rata-rata *offset* pada pengujian pertama adalah $1,89^\circ$. Pengujian kedua terjadi *steady state* sehingga *error* rata-rata setelah *rise time* $7,89^\circ$. Pengujian ketiga rata-rata *error offset* adalah $2,98^\circ$. Gangguan putaran searah jarum jam terjadi jika nilai aktual kompas melebihi *setpoint* dan sebaliknya gangguan putaran berlawanan jarum jam *quadcopter* terjadi jika nilai aktual kompas kurang dari *setpoint*.

4. Kesimpulan

Pengujian GPS U-blox Neo M8N didapatkan *error* terbesar 6,57 Meter, *error* terkecil 1,33 Meter dan *error*

rata-rata 3,86 Meter sehingga navigasi *bearing quadcopter* efektif digunakan dalam radius lebih dari 6,57 Meter dari titik tujuan. Pengujian HMC5883L didapatkan *error* terbesar mencapai 22° , setelah melalui proses perbaikan nilai *error* terbesar 6° dan rata-rata *error* $3,4^{\circ}$. Sistem kontrol navigasi *bearing* pada *quadcopter* dengan kontrol PID *self tuning* PSO, hasil pengujian statis didapatkan rata-rata *error offset* $5,4^{\circ}$ dan memiliki *settling time* pada detik ke 6,4. Pengujian dinamis memiliki *error offset* rata-rata $7,9^{\circ}$. Pengujian diberi gangguan didapatkan *error offset* rata-rata $1,89^{\circ}$ dan mencapai *rise time* pada detik ke 4,1. Nilai batasan para yang memiliki *rise time* yang cepat, *overshoot* yang rendah dan stabil adalah $K_p = 0,15$ sampai $0,3$, $K_i = 0,06$ sampai $0,6$ dan $K_d = 0,005$ sampai $K_d = 0,1$. Nilai koefisien PSO didapatkan melalui pengujian empiris, dari hasil pengujian empiris didapatkan nilai koefisien terbaik $C_1 = 1,5$ dan $C_2 = 2$. Untuk nilai koefisien bobot inersia terbaik dari $0,7$ sampai $1,2$.

Referensi

- [1]. M. Rizky Wiguna Utama, M. Komarudin, and A. Trisanto, *Sistem Kendali Holding Position Pada Quadcopter Berbasis Mikrokontroler Atmega 328p*. Bandar Lampung: Jurusan Teknik Elektro Universitas Lampung, Bandar Lampung, 2013.
- [2]. S. Zulu, Andrew., John, "A Review of Control Algorithms for Autonomous Quadrotors," *Open J. Appl. Sci.* 2014, 4, 547-556, 2014.
- [3]. H. D. Purnomo, *Cara Mudah Belajar Metode Optimisasi Metaheuristik Menggunakan Matlab*. Yogyakarta, 2014.
- [4]. S. S. Rao, *Engineering Optimization, Theory and Practice*, Fourth ed. New York, 2009.
- [5]. Y. S. and R. C. Eberhar, "Parameter selection in particle swarm optimization," in *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, 1998.
- [6]. E. Dermawan, A. Firmanto, S. N. Ambo, E. Diniardi, and A. I. Ramadhan, "Desain Frekuensi Kontrol pada Hybrid Wind Diesel dengan PID Particle Swarm Optimization (PSO)," *Semin. Nas. Sains dan Teknol. 2016 Fak. Tek. Univ. Muhammadiyah Jakarta*, 2016.
- [7]. Micromega Corporation, "Reading GPS Data," 2013. [Online]. Available: <http://www.micromegacorp.com/downloads/appnotes/A-N101-Reading-GPS-data.pdf>. [Accessed: 15-Aug-2017].
- [8]. Honeywell, "3-Axis Digital Compass IC HMC5883L." [Online]. Available: <https://media.digikey.com/PDF/Data-Sheets/Honeywell-PDFs/HMC5883L.pdf>. [Accessed: 08-Oct-2017].
- [9]. D. Luknanto, *Metode Numerik*. Yogyakarta: Bahan Kuliah Metode Numerik Jurusan Teknik Sipil FT UGM, 2001.