

IMPLEMENTASI PENGENDALI PID UNTUK NAVIGASI AUTONOMOUS BERBASIS GLOBAL POSITIONING SYSTEM PADA PURWARUPA AUTONOMOUS SURFACE VEHICLE

Alif Ihza Ahmada^{*)}, Wahyudi dan Eko Handoyo

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

^{*)E-mail : alifahmada@students.undip.ac.id}

Abstrak

Indonesia merupakan negara kepulauan yang memiliki wilayah perairan yang sangat luas, sehingga wilayah perairan yang harus dijaga juga sangat luas. Pada penelitian ini dirancang purwarupa sistem *autonomous surface vehicle* (ASV) yang kedepannya dapat digunakan untuk pengawasan wilayah perairan Indonesia. Perancangan sistem ASV meliputi perancangan perangkat keras yang digunakan dan perancangan pengendalian secara *autonomous* dengan pengendali Proporsional Integral Derifatif (PID) untuk mengendalikan motor servo. Perangkat keras yang digunakan untuk navigasi secara *autonomous* meliputi GPS receiver dan sensor magnetometer LIS3MDL. Sensor magnetometer digunakan sebagai umpan balik pengendali PID dan GPS receiver digunakan untuk mengetahui posisi dari ASV secara *real time*. Navigasi *autonomous* dilakukan dengan memberikan perintah berupa *waypoint* ke ASV. Pengendali PID akan membuat ASV mengikuti *waypoint* yang diberikan secara *autonomous*. Perancangan Pengendali PID dengan metode *tuning trial and error* didapatkan parameter K_p 1,5, K_i 0,3, dan K_d 0,2. Hasil pengujian PID yang dilakukan menunjukkan ASV dapat mencapai *setpoint* dalam waktu rata-rata selama 4,06 detik. Pengujian sistem kendali PID pada ASV dilakukan dengan beberapa bentuk lintasan, yaitu zigzag, bujur sangkar, dan sembarang. ASV dapat menyelesaikan misi pada setiap lintasan yang diberikan.

Kata kunci: Autonomous Surface Vehicle, ASV, PID, Waypoint.

Abstract

Indonesia is an archipelago country with vast water territory, so that the water territory that need to be guarded is very large. The purpose of this research is to design the hardware that needed for autonomous surface vehicle and to design autonomous control system with PID control for controlling servo motor. Hardware that used in autonomous navigation are GPS receiver and LIS3MDL magnetometer sensor. The magnetometer sensor used as feedback for the PID control and the GPS receiver used to locate the ASV in real time. Autonomous avigation done by giving instruction consist of waypoint to the ASV. The PID control will guide the ASV to follow the given waypoint autonomously. The PID design with trial and error tuning method resulting value of K_p 1.5, K_i 0.3, and K_d 0.2. The tested PID control showed that the ASV could reach the desired setpoint with an average time of 4.06 seconds. PID control tested in several waypoint shape, such as zigzag, rectangular, and random. The ASV could finish the given mission on every waypoint shape.

Keywords: Autonomous Surface Vehicle, ASV, PID, Waypoint.

1. Pendahuluan

Indonesia merupakan negara dengan perairan lebih luas dari daratannya. Seiring berkembangnya waktu, kondisi perairan mulai memburuk. Bahan bakar minyak merupakan salah satu penyebab dari terjadinya perubahan iklim, begitu juga dengan tingginya pertumbuhan penduduk. Indonesia membutuhkan suatu moda transportasi atau sebuah alat yang mampu menjaga dan mengeksplor perairan Indonesia dengan memperhatikan dampak terhadap lingkungan.

Autonomous Underwater Vehicle (AUV) telah banyak digunakan untuk mempelajari sumberdaya di lautan. Untuk meningkatkan efisiensi dari penelitian menggunakan AUV, *Autonomous Surface Vehicle* (ASV) telah dikembangkan. Untuk menunjang aktifitas AUV, sebuah sistem jelajah rute dibutuhkan di ASV [1]. *Autonomous Surface Vehicle* (ASV) adalah kendaraan robotik yang beroperasi pada permukaan laut [2]. Kapal dengan teknologi *autonomous* akan membantu dalam pengawasan wilayah perairan. Dengan kemampuan *autonomous* kapal dapat menyusuri wilayah perairan yang perlu untuk diawasi dan dipantau secara *real time*. Dengan kapal *autonomous*, perairan yang berbahaya untuk dijangkau

manusia akan dapat dijangkau dengan menggunakan kapal. Perkembangan ASV di dunia sudah pesat. Saat ini, survei batimetri dan oseanografi dapat dilakukan dengan menggunakan ASV, seperti Delfim, Sesamo, IRIS, SCOUT, dan ROAZ. ASV memiliki manfaat yang cukup besar dalam observasi perairan. Namun perkembangan ini belum diikuti dengan baik di Indonesia, sehingga perlu dilakukan pengembangan lebih lanjut pada teknologi ASV [3].

Pada penelitian ini dirancang perangkat keras yang dibutuhkan untuk navigasi pada purwarupa ASV dengan menggunakan navigasi GPS. Navigasi menggunakan *waypoint* yang dikirimkan dari *Ground Control Station* (GCS). Pengendalian yang digunakan untuk navigasi pada ASV adalah pengendali PID yang menggerakkan *rudder* pada kapal. Pengendali proporsional digunakan untuk memperbaiki respon transien khususnya *rise time* dan *settling time*. Keluaran dari pengendali proporsional sebanding dengan besarnya sinyal *error*, pengendali integral untuk menghilangkan *steady state error*, dan pengendali derivatif untuk memperbaiki stabilitas sistem [4].

Pada penelitian digunakan persamaan Vincenty yang digunakan untuk menghitung jarak antara dua buah titik pada permukaan bumi dengan asumsi bahwa bentuk bumi adalah bulat tidak sempurna [5]. Setpoint pengendali PID menggunakan persamaan Vincenty digunakan untuk mengendalikan motor servo.

2. Metode

2.1. Perancangan Perangkat Keras

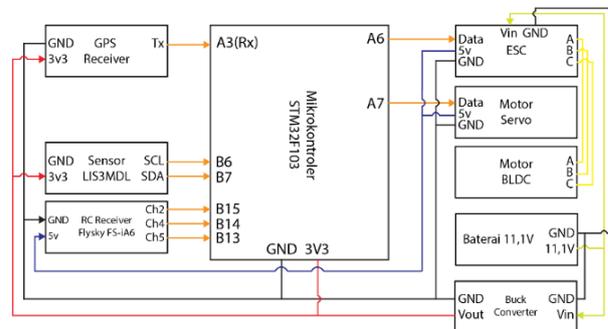
Perangkat keras yang digunakan berupa mikrokontroler STM32F103, sensor magnetometer LIS3MDL, GPS receiver, radio telemetry, dan aktuator berupa motor servo dan motor BLDC. Mikrokontroler berfungsi sebagai pengolah data dari sensor-sensor yang digunakan dan digunakan untuk melakukan pengontrolan pada aktuator.

2.1.1. Hubungan Komponen Pendukung dengan Mikrokontroler

Mikrokontroler STM32F103 berfungsi sebagai pengolah data sensor LIS3MDL dan GPS receiver, kemudian data yang diterima akan diolah untuk digunakan pada pengendalian motor servo. Pin pada mikrokontroler yang digunakan adalah pin I2C, pin serial, dan pin PWM. Pin I2C dihubungkan dengan sensor magnetometer LIS3MDL, pin serial dihubungkan dengan GPS receiver dan modul telemetry, dan pin PWM dihubungkan dengan motor BLDC dan motor servo.

Mikrokontroler STM32F103 bekerja pada tegangan 3,3 V, sumber tegangan berasal dari baterai LiPo 11,1 V yang diturunkan tegangannya dengan regulator *buck converter* menjadi 3,3 V. Motor servo dan motor BLDC memerlukan

tegangan 5 V, dengan regulator *step-down* tegangan baterai LiPo dari 11,1 V diturunkan menjadi 5 V.



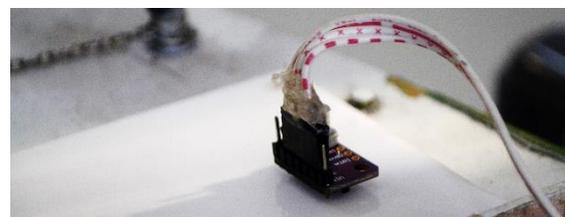
Gambar 1. Penggunaan pin mikrokontroler

2.1.2. Sensor LIS3MDL

Sensor magnetometer LIS3MD adalah sensor yang berfungsi untuk membaca besar medan magnet bumi. Sensor ini beroperasi pada tegangan 1,9 V hingga 3,6 V. Sensor LIS3MD dapat diakses dengan menggunakan komunikasi I2C dengan alamat 0x1E [6].

Sensor LIS3MDL bekerja pada tegangan 3,3 V dan menggunakan sumber tegangan yang berasal dari baterai yang diturunkan tegangannya menggunakan *buck converter*. Sensor LIS3MDL menggunakan protokol komunikasi I2C dalam pengiriman data ke mikrokontroler. Pin SDA dan SCL sensor dihubungkan ke pin SDA (pin B7) dan pin SCL (pin B6) pada mikrokontroler.

Pada badan kapal sensor LIS3MDL diletakkan pada bagian depan untuk menghindari gangguan dari motor BLDC maupun motor servo. Peletakan sensor LIS3MDL penting karena sensor magnetometer rentan terhadap gangguan oleh logam dan medan magnet. Posisi sensor seperti pada Gambar 2.



Gambar 2. Posisi sensor LIS3MDL pada ASV

2.1.3. GPS Receiver

Global Navigation Satellite System (GNSS) adalah sebuah sistem navigasi satelit di mana satelit memberikan informasi geografis secara *autonomous*. Sistem GNSS terdiri dari beberapa sistem antara lain GPS yang dimiliki oleh Amerika Serikat, GLONASS milik Rusia, dan Galileo

milik Uni Eropa [4]. GPS menggunakan sebuah penerima elektronik untuk menentukan lokasi menggunakan sinyal waktu yang ditransmisikan melalui sinyal radio line of sight dari satelit [5]. GPS receiver akan membandingkan waktu sinyal yang dikirim dengan waktu sinyal diterima, sehingga diketahui jarak antara setiap satelit dengan GPS receiver. Dari perbandingan tersebut akan didapatkan titik koordinat dari GPS receiver. GPS receiver memerlukan minimal 3 sinyal satelit untuk mendapatkan titik koordinat dan untuk aplikasi dengan kebutuhan keselamatan tinggi memerlukan minimal 6 satelit [7].

GPS receiver yang digunakan adalah Ublox Neo 7M yang menggunakan tegangan kerja 3,3 V dan menggunakan protokol komunikasi serial asinkron. Untuk dapat mengakses data dari GPS, pin TX GPS dihubungkan dengan pin RX pada mikrokontroler dan pin RX GPS dihubungkan dengan pin Tx pada mikrokontroler. Keluaran data GPS berupa paket data dalam format NMEA, untuk keperluan navigasi pada ASV yang dibutuhkan adalah data GGA yang berisi data *latitude* dan *longitude*.

Untuk mendapatkan sinyal dari satelit, GPS receiver perlu ditempatkan pada posisi yang tidak tertutup. Pada kapal GPS receiver diletakkan dibagian depan dan antena GPS receiver dihadapkan ke atas seperti ditunjukkan pada Gambar 3.



Gambar 3. GPS Receiver pada ASV

2.1.4. Motor Servo

Motor servo adalah sebuah aktuator yang memungkinkan untuk dikendalikan posisi sudut, kecepatan, dan percepatannya. Pada sebuah motor servo terdapat sebuah motor DC yang di-couple dengan resistor yang digunakan sebagai umpan balik posisi. Pada motor servo juga terdapat rangkaian gear dan juga control circuit. Rangkaian gear berfungsi untuk memperlambat putaran poros dan meningkatkan torsi motor servo [8].

Motor servo yang digunakan pada penelitian ini adalah RotorStar yang memiliki tegangan kerja 5 V sampai 6 V dan memerlukan arus minimal 1 A, oleh karena itu motor servo perlu sumber tegangan yang terpisah dari mikrokontroler yang menggunakan tegangan kerja 3,3 V. Sumber tegangan untuk motor servo didapat dari baterai LiPo 11,1 V yang diturunkan menjadi 5 V menggunakan ESC yang sekaligus digunakan untuk motor BLDC. Motor servo memerlukan sinyal PWM untuk bisa bekerja. Pin

pada mikrokontroler yang digunakan adalah pin khusus yang dapat mengeluarkan sinyal PWM, yaitu pin A6 yang dihubungkan dengan pin sinyal pada motor servo.

Pemasangan motor servo pada hull kapal ditunjukkan pada Gambar 4. Motor servo dihubungkan dengan rudder yang digunakan kapal untuk bermanuver.



Gambar 4. Pemasangan motor servo pada kapal

2.1.5. Motor BLDC

Motor BLDC adalah motor yang tidak menggunakan brush pada komutatornya. Motor BLDC termasuk motor sinkron, artinya medan magnet yang dihasilkan stator dan medan magnet yang dihasilkan rotor berputar pada frekuensi yang sama [9]. Cara Kerja motor BLDC adalah dengan memberikan arus 3 fasa pada stator. Arus yang lewat menyebabkan munculnya medan magnet pada stator. Arus yang masuk ke motor merupakan arus AC 3 fasa sinusoidal, sehingga nilai medan magnet dan polarisasi dari setiap kumparan akan berubah-ubah setiap saat. Perubahan medan magnet dan polarisasi pada tiap kumparan ini menciptakan medan putar stator. [10].

Motor BLDC dapat dikontrol dengan mikrokontroler dengan disambungkan ke ESC dengan masukan berupa sinyal PWM. ESC memiliki 3 pin yang dihubungkan ke motor BLDC dan sebuah pin sinyal yang dihubungkan ke pin PWM pada mikrokontroler. Untuk mengendalikan motor BLDC pin A7 pada mikrokontroler dihubungkan pada pin sinyal ESC. ESC yang digunakan adalah Turnigy yang dapat bekerja pada tegangan 6-14 V, jadi ESC dapat dihubungkan langsung dengan baterai LiPo sebagai sumber tegangan.

2.1.6. RC Receiver

RC receiver adalah perangkat yang digunakan untuk menerima sinyal dari remote control. RC receiver yang digunakan adalah FlySky FS-iA6 yang menggunakan frekuensi kerja 2,4 GHz. Sinyal keluaran dari RC receiver adalah sinyal PWM dengan lebar pulsa antara 1 ms hingga 2 ms yang dapat dibaca oleh mikrokontroler [11].

RC receiver menerima sinyal dari remote control kemudian dibaca pada mikrokontroler dalam bentuk sinyal

PWM. Channel yang digunakan dari RC receiver sebanyak 3 buah, yaitu channel 2, channel 4, dan channel 5 yang masing-masing channel dihubungkan pada pin B15, pin B14, dan pin B13 pada mikrokontroler. Sinyal keluaran RC receiver dibaca dengan fungsi pulseIn dan setiap nilai PWM dari masing-masing channel disimpan dalam variabel rcValue[0], rcValue[1], dan rcValue[2]. Fungsi masing-masing channel adalah sebagai berikut.

1. Channel 2 digunakan untuk menggerakkan throttle motor BLDC.
2. Channel 4 digunakan untuk menggerakkan motor servo.
3. Channel 5 digunakan untuk tombol emergency.

2.2. Penentuan Setpoint

Setpoint untuk PID yang digunakan pada navigasi kapal didapat dari megolah data latitude dan longitude GPS kapal dan data latitude dan longitude dari waypoint yang diberikan oleh GCS. Data waypoint dari GCS disimpan dalam variabel wpLat dan wpLon. Dengan menggunakan persamaan vincenty akan didapatkan data sudut azimuth dan jarak, setpoint yang digunakan dalam pengendali PID adalah sudut azimuth yang didapatkan dari persamaan vincenty.

Persamaan Vincenty adalah persamaan yang dikembangkan oleh Theoddeus Vincenty pada tahun 1975. Persamaan Vincenty digunakan untuk menghitung jarak antara dua buah titik pada permukaan bumi dengan asumsi bahwa bentuk bumi adalah bulat tidak sempurna [12].

Pada penelitian ini digunakan persamaan Vincenty metode inverse untuk menentukan jarak dan sudut azimuth antara kapal dengan target waypoint yang diberikan. Perhitungan jarak dan sudut azimuth dapat dilihat pada persamaan 1 sampai 16.

$$U_1 = \tan^{-1}(1 - f)\tan\phi_1 \quad (1)$$

$$U_2 = \tan^{-1}(1 - f)\tan\phi_2 \quad (2)$$

$$\sin \sigma = \frac{\sqrt{(\cos U_2 \sin \lambda)^2 + (\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \lambda)^2}}{\sqrt{(\cos U_2 \sin \lambda)^2 + (\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \lambda)^2}} \quad (3)$$

$$\cos \sigma = \sin U_1 \sin U_2 + \cos U_1 \cos U_2 \lambda \quad (4)$$

$$\tan \sigma = \frac{\sin \sigma}{\cos \sigma} \quad (5)$$

$$\sin \alpha = \frac{\cos U_1 \cos U_2 \lambda}{\sin \sigma} \quad (6)$$

$$\begin{aligned} \cos(2\sigma_m) &= \cos \sigma - \frac{2 \sin U_1 \sin U_2}{\cos^2 \alpha} \\ &= \cos \sigma - \frac{2 \sin U_1 \sin U_2}{1 - \sin^2 \alpha} \end{aligned} \quad (7)$$

$$C = \frac{f}{16} \cos^2 \alpha [4 + f(4 - 3 \cos^2 \alpha)] \quad (8)$$

$$\begin{aligned} \lambda &= L + (1 - C) f \sin \alpha \{ \sigma + C \sin \sigma [\cos(2\sigma_m) + \\ &C \cos \sigma (-1 + 2 \cos^2(2\sigma_m))] \} \end{aligned} \quad (9)$$

Dari persamaan 3 hingga persamaan 7 didapatkan nilai λ , perhitungan tersebut diulangi hingga perubahan pada λ dapat diabaikan. Setelah didapatkan nilai λ , dilakukan

perhitungan untuk mencari jarak dan sudut azimuth dengan persamaan 10 hingga 16.

$$u^2 = \cos^2 \alpha \left(\frac{a^2 - b^2}{b^2} \right) \quad (10)$$

$$A = 1 + \frac{u^2}{16384} (4096 + u^2 [-768 + u^2 (320 - 175u^2)]) \quad (11)$$

$$B = \frac{u^2}{1024} (256 + u^2 [-128 + u^2 (74 - 47u^2)]) \quad (12)$$

$$\begin{aligned} \Delta \sigma &= B \sin \sigma \left\{ \cos(2\sigma_m) + \frac{1}{4} B \left(\cos \sigma [-1 + \right. \right. \\ &2 \cos^2(2\sigma_m)] - \frac{B}{6} \cos[2\sigma_m] [-3 + 4 \sin^2 \sigma] [-3 + \\ &4 \cos^2(2\sigma_m)] \left. \left. \right\} \end{aligned} \quad (13)$$

$$s = bA(\sigma - \Delta \sigma) \quad (14)$$

$$\alpha_1 = \tan^{-1} \left(\frac{\cos U_2 \sin \lambda}{\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos \lambda} \right) \quad (15)$$

$$\alpha_2 = \tan^{-1} \left(\frac{\cos U_2 \sin \lambda}{-\sin U_1 \cos U_2 + \cos U_1 \sin U_2 \cos \lambda} \right) \quad (16)$$

dengan

a = jari-jari bumi pada khatulistiwa (6378137.0 m)

f = flattening of ellipsoid (1/298.257223563)

b = jari-jari bumi dari kutub (6356752.314245 m)

ϕ = latitude dari titik koordinat

L_1, L_2 = longitude dari titik koordinat

L = selisih longitude dari 2 titik koordinat ($L_2 - L_1$)

λ = selisih longitude titik pada ruang bantu

α_1, α_2 = azimuth pada titik koordinat

α = azimuth dari geodesi pada khatulistiwa

S = jarak dari dua titik koordinat

σ_m = sudut pemisah antara titik tengah pada garis dan khatulistiwa

Dari persamaan 14 didapatkan jarak antara 2 titik koordinat, dan pada persamaan 16 didapatkan sudut azimuth terhadap utara dari titik koordinat pertama ke titik koordinat kedua.

2.3. Perancangan Pengendali PID

Pengendali PID adalah sebuah pengendali yang menggabungkan antara pengendali proporsional ditambah pengendali integral ditambah pengendali derivative, persamaan PID dapat dilihat pada persamaan 17.

$$u(t) = K_p e(t) + K_i \int_0^t e(t') dt' + K_d \frac{de(t)}{dt} \quad (17)$$

Untuk dapat diimplementasikan pada mikrokontroler persamaan pengendali PID yang digunakan adalah persamaan PID dalam bentuk diskret. Pengendali proporsional dalam bentuk diskret ditunjukkan pada persamaan 18. Pengendali integral dan derivatif, bentuk diskret diperoleh dengan menggunakan pendekatan backward differentiation sehingga didapatkan persamaan 19. dan persamaan 20 [13].

$$u_p(k) = K_p e(k) \quad (18)$$

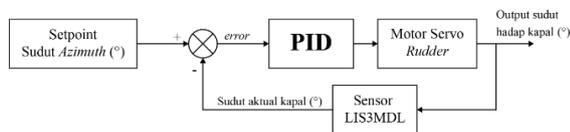
$$u(k) = K_i \sum_{i=0}^k e(k) T_s \quad (19)$$

$$u_d(k) = \frac{K_d}{T_s} (e(k) - e(k - 1)) \quad (20)$$

Persamaan PID diskret yang digunakan dalam mikrokontroler seperti pada persamaan 21.

$$u(k) = K_p e(k) + K_i \sum_{i=0}^k e(k) T_s + \frac{K_d}{T_s} (e(k) - e(k - 1)) \quad (21)$$

Pada penelitian ini pengendali PID digunakan sebagai pengendali motor servo *rudder* yang digunakan untuk navigasi secara *autonomous*. Pengendali PID diimplementasikan dalam bentuk senarai program yang ditanamkan pada mikrokontroler. Nilai *sampling time* (T_s) yang digunakan berdasarkan waktu interval proses *looping* pada mikrokontroler yaitu 0,02 detik yang didapat dengan menggunakan fungsi `millis()` pada arduino. Pengendali PID menggunakan input berupa sinyal *error* yang didapat dari selisih antara *setpoint* dan sudut *heading* ASV. *Setpoint* yang digunakan adalah sudut *azimuth* dan feedback dari sistem adalah sudut hadap ASV yang dibaca menggunakan sensor LIS3MDL. Keluaran dari pengendali PID adalah nilai PWM yang digunakan sebagai inputan motor servo *rudder* untuk navigasi secara *autonomous* dan hasil akhir dari keluaran sistem adalah arah hadap kapal. Blok diagram pengendali PID dapat dilihat pada Gambar 5.



Gambar 5. Blok diagram pengendali PID

Pada penelitian ini metode *tuning* yang dilakukan adalah *tuning trial and error* yang dilakukan berdasarkan karakteristik pengendali PID. Dari hasil *tuning* parameter K_p , K_i , dan K_d didapatkan nilai K_p sebesar 1,5, nilai K_i sebesar 0,3, dan nilai K_d sebesar 0,2.

Keluaran pengendali PID dibatasi pada rentang -675 hingga 675, angka ini didapat dari mengurangi batas maksimal pwm servo 2175 μ s dengan titik tengah pwm servo yaitu 1500 μ s. Saat keluaran PID bernilai positif ASV akan berbelok ke kiri dan saat keluaran PID bernilai negatif ASV akan berbelok ke kanan, sehingga ASV dapat bergerak pada permukaan air secara otomatis dari satu titik ke titik lain dengan bantuan waypoint yang telah ditentukan sebelumnya [14].

3. Hasil dan Analisis

3.1. Pengujian sensor LIS3MDL

Pengujian sensor dilakukan dengan membandingkan nilai *heading* pembacaan sensor dengan sudut referensi pada

busur yang pada sudut 0° sudah disesuaikan dengan arah utara menggunakan kompas pada smartphone.

Sebelum dilakukan pengujian sensor telah dikalibrasi dengan cara mencari nilai maksimal dan minimal masing-masing sumbu x dan y. Data sudut hasil pengujian dibandingkan dengan sudut referensi untuk mendapatkan nilai *error* dari sensor. Hasil pengujian sebelum kalibrasi sensor memiliki rerata *absolute error* sebesar 15,6° dengan *absolute error* terbesar sebesar 34°. Setelah dilakukan kalibrasi, pengujian menunjukkan sensor memiliki rerata *absolute error* sebesar 3,4° dengan *absolute error* terbesar sebesar 9°.

3.2. Pengujian GPS Receiver

GPS *receiver* diuji untuk mengetahui akurasi pembacaan lokasi dari *receiver*. Pengujian dilakukan dengan membandingkan data lokasi dari GPS dengan data lokasi dari batu penanda (*benchmark*) koordinat lokasi yang ada di Universitas Diponegoro. GPS menggunakan protokol komunikasi serial asinkron yang biasa disebut UART. Komunikasi UART menggunakan dua buah pin untuk mengirim dan menerima data, yaitu pin TXD dan RXD. Pin TX berfungsi untuk mengirim data dan RX untuk menerima data [15].

Pengujian dilakukan dengan cara meletakkan GPS *receiver* di atas *benchmark* dengan mengambil data selama 10 detik, data dari GPS yang digunakan adalah data *latitude* dan *longitude*. Nilai *error* pengujian GPS didapatkan dengan membandingkan koordinat keluaran GPS dengan koordinat *benchmark* sebagai referensi. Pengujian GPS dilakukan pada 3 *benchmark* berbeda, nilai *latitude* dan *longitude* dari setiap *benchmark* adalah sebagai berikut :

1. Jalan sebelah barat Polines (-7,05166734; 110,43305556).
2. Depan koperasi Undip (-7,05367950; 110,43944444).
3. Depan Student Center Undip (-7,05340466; 110,43861111).

Dari pengujian yang dilakukan didapatkan nilai *error* jarak GPS pada *benchmark* 1 sebesar 3,1 m, pada *benchmark* 2 sebesar 3,4 m, pada *benchmark* 3 sebesar 3 m, dengan rata-rata *error* sebesar 3,16 m. Nilai *error* hasil pengujian masih dalam rentang *error* sesuai spesifikasi berdasarkan *datasheet*, sehingga GPS *receiver* dapat digunakan pada ASV.

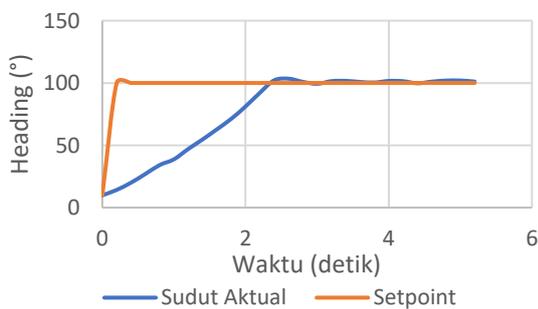
3.3. Pengujian Pengendali PID

Pengujian pengendali PID dilakukan untuk mengetahui respon sistem saat diberikan pengendali PID yang sudah di-*tuning* dengan metode *trial and error*. Hasil *tuning trial and error* yang dilakukan didapatkan nilai parameter K_p 1,5, K_i 0,3, dan K_d 0,2. Pengujian dilakukan dengan memberikan *setpoint* berupa sudut *azimuth*. Parameter yang diambil dari pengujian yaitu *settling time* dari respon

sistem. Hasil dari pengujian pengendali PID kemudian digunakan untuk pengujian sistem secara *autonomous* dengan *waypoint*. Pengujian dilakukan dengan 3 variasi *setpoint*, yaitu 100° , 190° , dan 270° . *Setpoint* yang digunakan dalam pengujian *tuning* ini merupakan *setpoint* yang dipakai pada pengujian dengan *waypoint*.

A. *Setpoint* 100°

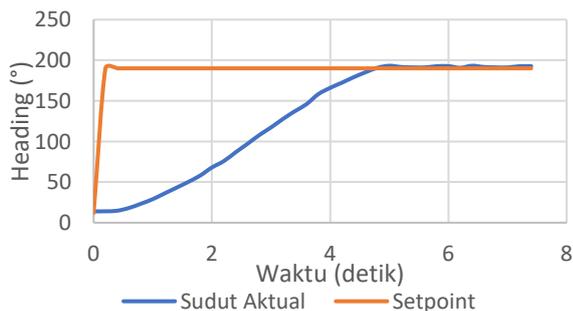
Hasil pengujian pengendali PID pada *setpoint* 100° ditampilkan pada Gambar 6. Sudut *heading* awal pengujian dipilih sudut 10° agar grafik yang ditampilkan terlihat jelas. Pada gambar terlihat respon naik mulai dari sudut 10° hingga mencapai *setpoint* dalam waktu 3,6 detik.



Gambar 6. Respon sistem pada *setpoint* 100°

B. *Setpoint* 190°

Hasil pengujian pengendali PID pada *setpoint* 190° ditampilkan pada Gambar 7. Sudut *heading* awal pengujian dipilih sudut 10° agar grafik yang ditampilkan terlihat jelas. Pada gambar terlihat respon naik mulai dari sudut 10° hingga mencapai *setpoint* dalam waktu 4,8 detik.

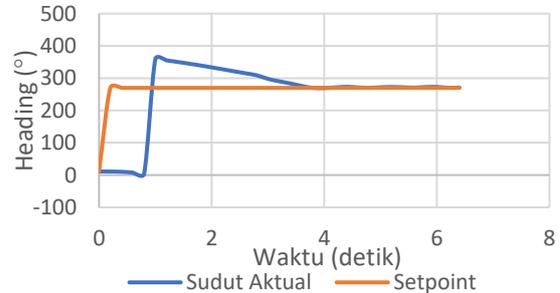


Gambar 7. Respon sistem pada *setpoint* 190°

C. *Setpoint* 270°

Hasil pengujian pengendali PID pada *setpoint* 270° ditampilkan pada Gambar 8. Sudut *heading* awal pengujian dipilih sudut 10° agar grafik yang ditampilkan terlihat jelas. Pada gambar terlihat respon naik mulai dari sudut 10° hingga mencapai *setpoint* dalam waktu 3,8 detik.

Berdasarkan hasil pengujian pengendali PID yang dilakukan terdapat selisih waktu untuk sistem mencapai *setpoint* dengan rata-rata waktu *settling time* selama 4,06 detik.



Gambar 8. Respon sistem pada *setpoint* 270°

Tabel 1. Hasil pengujian PID

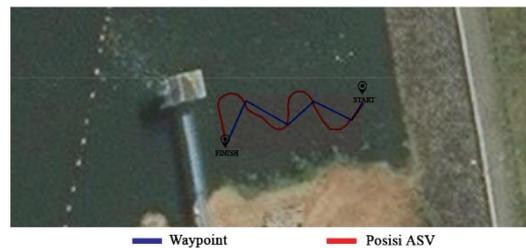
No.	Setpoint(°)	Settling Time (s)
1	100	3,6
2	190	4,8
3	270	3,8
Rata-rata		4,06

3.4. Pengujian Sistem *Autonomous* dengan *Waypoint*

Pengujian sistem *autonomous* dilakukan dengan mengirimkan *waypoint* ke ASV. Sistem navigasi *autonomous* dikatakan berhasil apabila ASV mampu menuju *waypoint* yang diberikan. Pengujian sistem *autonomous* dilakukan di Waduk Pendidikan Universitas Diponegoro. Pengujian diberikan tiga variasi *waypoint* yang dilalui ASV, pertama *waypoint zigzag*, *waypoint* bujur sangkar, dan *waypoint* sembarang.

A. *Waypoint zigzag*

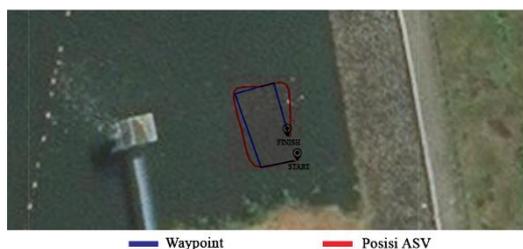
Pada pengujian dengan *waypoint zigzag* ASV dapat menuju *waypoint* yang diberikan. Waktu yang dibutuhkan ASV dari titik awal sampai titik terakhir *waypoint* dengan panjang lintasan 34,3 meter adalah 40,2 detik. ASV dapat berbelok menuju *waypoint* dengan sedikit melebar dari jalur *waypoint* yang diberikan, hal ini karena ASV memerlukan waktu selama 4 detik untuk dapat menghadap ke *setpoint* sudut *azimuth* seperti yang diperlihatkan pada Gambar 9.



Gambar 9. Pengujian *waypoint zig-zag* dengan PWM motor 1375ms

B. Waypoint bujur sangkar

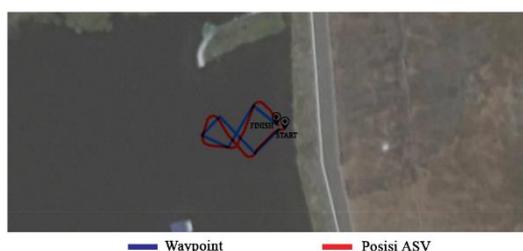
Pada pengujian dengan *waypoint* bujur sangkar ASV dapat menuju *waypoint* yang diberikan. Waktu yang dibutuhkan ASV dari titik awal sampai titik terakhir *waypoint* dengan panjang lintasan 40,51 meter adalah 35,6 detik. ASV dapat berbelok menuju *waypoint* dengan sedikit melebar dari jalur *waypoint* yang diberikan, hal ini karena ASV memerlukan waktu selama 2 detik untuk dapat menghadap ke *setpoint* sudut *azimuth* seperti yang diperlihatkan pada Gambar 10.



Gambar 10. Pengujian *waypoint* bujur sangkar dengan PWM motor 1375ms

C. Waypoint sembarang

Pada pengujian dengan *waypoint* sembarang ASV dapat menuju *waypoint* yang diberikan. Waktu yang dibutuhkan ASV dari titik awal sampai titik terakhir *waypoint* dengan panjang lintasan 70,55 meter adalah 60,8 detik. ASV dapat berbelok menuju *waypoint* dengan sedikit melebar dari jalur *waypoint* yang diberikan, hal ini karena ASV memerlukan waktu selama 1,8 detik untuk dapat menghadap ke *setpoint* sudut *azimuth* seperti yang diperlihatkan pada Gambar 11.



Gambar 11. Pengujian *waypoint* sembarang dengan PWM motor 1375ms

Dari hasil pengujian dengan *waypoint* yang berbeda didapatkan ASV dapat menyelesaikan misi dengan waktu rata-rata 34,8 detik. Pada pengujian dengan *waypoint zigzag* ASV memiliki respon lebih lama dibandingkan dengan *waypoint* yang lain, hal ini menunjukkan bahwa waktu yang diperlukan ASV untuk mencapai *setpoint* dipengaruhi oleh bentuk dari *waypoint* yang diberikan.

4. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, maka dapat disimpulkan beberapa hal. Pengujian sensor magnetometer LIS3MDL memiliki *error* rata-rata sebesar $3,4^\circ$. GPS *receiver* Ublox Neo-7M memiliki *error* rata-rata sebesar 3,16 meter. Dari *tuning trial and error* didapatkan nilai K_p 1,5, K_i 0,3, K_d 0,2. Pengujian pengendali PID menunjukkan ASV dapat mencapai *settling time* dalam waktu rata-rata selama 4,06 detik. Waktu yang diperlukan ASV untuk mencapai titik terakhir *waypoint* dipengaruhi oleh bentuk *waypoint*.

Referensi

- [1]. Kanako Kobatake, Tadatsugi Okazaki, Masakazu Arima, "Study on Optimal Tuning of PID Autopilot for Autonomous Surface Vehicle," International Federation of Automatic Control, 2019.
- [2]. "Autonomous Surface Vehicles," [Online]. Tersedia: <https://www.noc.ac.uk/facilities/marine-autonomous-robotic-systems/asv>. Diakses: 28 April 2020.
- [3]. Mahesa G. A. Satriya, Indra Jaya, Yopi Novita, "Integrasi Rancangan Sistem Observasi Kapal Permukaan Otomatis dengan Google Earth," Institut Pertanian Bogor, 2016.
- [4]. Araki M, "PID Control", Control System, Robotics, And Automation - Vol. II. hal. 1-6.
- [5]. T. Vincenty, "Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations," Directorate of Overseas Surveys, Wyoming, 1975.
- [6]. LIS3MDL, STMicroelectronics, 2017.
- [7]. A Beginner's Guide to GNSS in Europe, International Federation of Air Traffic Controllers Associations, 1999.
- [8]. Ahmad Riyandi, "Aplikasi Logika Fuzzy Sebagai Auto Tuning Parameter Kontroler PID pada Pengendalian Antena Tracker Berbasis GPS," Laporan Tugas Akhir, Departemen Teknik Elektro, Universitas Diponegoro, Semarang, 2018.
- [9]. Brushless DC Motor Fundamentals, Microchip Technology Inc., 2003.
- [10]. Mega Rosaliana, "Perancangan Kontroler Self-Tuning PID Dengan Logika Fuzzy Sebagai Kedali Kecepatan Motor Brushless DC (BLDC)," Laporan Tugas Akhir, Departemen Teknik Elektro, Universitas Diponegoro, 2018.
- [11]. FS-iA6 User Manual, FlySky RC model technology co., ltd, Shenzhen, 2016.
- [12]. Ellipsoid Oblate Spheroid Earth." [Online]. Tersedia: <https://gisgeography.com/ellipsoid-oblate-spheroid-earth/>. Diakses: 8 Mei 2020.
- [13]. Finn Haugen, "Discretization of simulator, filter, and PID controller," Article, Tech Teach, 2010.
- [14]. Fadlila Rizki Saputra, Muhammad Rivai, "Autonomous Surface Vehicle sebagai Alat Pemantau Lingkungan Menggunakan Metode Navigasi Waypoint," Jurnal Teknik ITS, 2018.
- [15]. USART Peripheral Interface, Texas Instruments, 2012.