

PERANCANGAN SISTEM KENDALI *MOBILE ROBOT* DENGAN GPS MENGUNAKAN METODE PID

CONTROL SYSTEM DESIGN OF MOBILE ROBOT WITH GPS USING PID METHOD

Rahardi Prakoso¹, Porman Pangaribuan², Agung Surya Wibowo³

^{1,2,3}Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

¹rahardiprakoso@yahoo.com ²porman@telkomuniversity.ac.id ³agungsw@telkomuniversity.ac.id

Abstrak

Mobile robot adalah konstruksi robot yang memiliki beberapa roda untuk menggerakkan robot tersebut, sehingga dapat berpindah dari satu posisi ke posisi yang lain. Untuk menggerakkan *mobile robot* dapat menggunakan banyak cara, salah satunya dengan *waypoint control*. *Waypoint control* adalah sebuah teknik dimana sebuah benda dapat berpindah secara otomatis, sesuai dengan perintah operator. Operator akan memberi nilai *input* (titik acuan) berupa titik *longitude* dan *latitude* ke sistem *mobile robot*. Kendali yang digunakan untuk mengolah data menggunakan kendali PD. Lalu hasil *output* dari PD akan berupa PWM (*Pulse Width Modulation*) yang akan digunakan untuk menggerakkan motor DC sebagai aktuator *mobile robot*. Rctimer NEO-6M GPS Module and MAG Compass v1.2 6M-MAG yang terdiri dari IC U-blox Neo 6m akan digunakan untuk membaca posisi aktual robot dan IC HMC5883L akan digunakan sebagai penunjuk arah orientasi robot. Posisi aktual robot akan terus dibaca guna menentukan jarak dan beda sudut antara *mobile robot* dengan titik tujuan. Hasil penelitian yang telah dilaksanakan menunjukkan bahwa *mobile robot* dapat mencapai titik acuan yang berjarak 20 m dengan waktu 42 detik.

Kata Kunci : PWM, *waypoint control*, *latitude*, *longitude*

Abstract

Mobile robot is a robot construction that has several wheels to move the whole robot, so it can move from one position to another. There are many ways to move the mobile robot, one of them is *waypoint control*. *Waypoint control* is a technique where an object can move automatically, according to the operator's command. The operator assigns the input value (point of reference) to the point of longitude and latitude to the mobile robot system. Full control used to process data using PD control. Later the output of PD is PWM (*Pulse Width Modulation*) which will be used to drive a DC motor as a mobile robot actuator. The RCTimer NEO-6M GPS Module and MAG Compass v1.2 6M-MAG comprising the U-blox Neo 6m IC will be used to read the actual position of the robot and the HMC5883L IC will be used as a robot orientation pointer. The actual position of the robot will continue to be read in order to determine the distance and the different angle between the mobile robot and the destination point. Based on results of the research have been carried out, mobile robot can reach the reference point which is 20 m with 42 seconds.

Keywords : PWM, *waypoint control*, *latitude*, *longitude*

1. Pendahuluan

Beberapa tahun terakhir telah dikembangkan, sistem navigasi yang membutuhkan ukuran kecil dan sumber daya yang terbatas pada kondisi diluar ruangan [1], maka dari itu perkembangan *mobile robot* cukup pesat. *Mobile robot* adalah konstruksi robot yang mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot, sehingga robot tersebut dapat berpindah[2]. Sudah banyak metode yang mempelajari tentang bagaimana sebuah *mobile robot* dapat bergerak. Namun sistem navigasi *waypoint control* belum banyak di teliti oleh sebagian orang. Sebenarnya metode ini sudah banyak diterapkan di robot air USV (*Unmanned Surface Vehicle*), tapi di *mobile robot* yang berjenis robot darat, sistem navigasi ini belum banyak diterapkan.

Waypoint control adalah sebuah sistem navigasi yang memungkinkan sebuah *plant* untuk bergerak secara otomatis berdasarkan *input* yang diberikan oleh operator. Dengan menerapkan sistem navigasi ke *mobile robot*, maka *mobile robot* akan bergerak secara otomatis ketika operator sudah memberikan nilai input. Nilai *input* yang akan diberikan oleh operator adalah titik koordinat. Terdapat 3 jenis titik koordinat yaitu, *latitude*, *longitude* dan *altitude*. Tapi dalam tugas akhir ini hanya akan menggunakan *latitude* dan *longitude* saja.

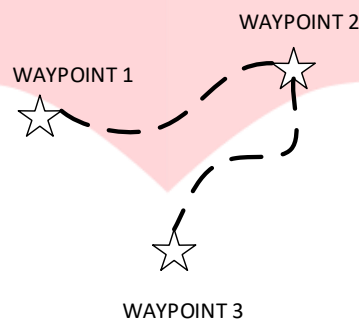
Pada penelitian ini, penulis akan merancang *mobile robot* yang dapat berpindah dari satu titik ke titik lainnya. *Mobile robot* ini akan menggunakan sistem kendali dengan teknik *waypoint* menggunakan metode PID. Pemilihan jenis kendali PID didasari atas PID merupakan pengontrol yang paling populer di bidang industri karena kesederhanaannya, ketangguhannya dan kinerjanya yang maksimal[3]. Kendali PID ini akan mengatur nilai *error* jarak dan *error* sudut. Keluaran dari PID berupa PWM (*pulse width modulation*) yang akan digunakan sebagai masukan

untuk aktuator (motor DC). Jumlah roda yang digunakan berjumlah dua buah roda aktif dibagian belakang robot dan satu roda pasif yang terletak dibagian depan robot.

2. Dasar Teori

2.1. Waypoint Control

Sistem navigasi pada *mobile robot* dapat diartikan sebagai suatu kemampuan untuk memandu pergerakan dari suatu posisi ke posisi lain yang dituju melalui penentuan posisi dan arah gerakannya[4]. Dalam pembuatan *mobile robot* ini, sistem navigasi yang digunakan adalah *waypoint control*. *Waypoint control* dapat diartikan dengan suatu sistem pergerakan titik dari koordinat titik awal terhadap titik acuan pada bidang XY[5]. Teknik ini dapat sangat bermanfaat jika ingin menuju tempat yang sulit dijangkau oleh manusia, karena dengan teknik ini operator hanya memasukkan titik acuan yang ingin dicapai oleh objek. Titik acuan yang digunakan dalam penelitian ini berupa titik koordinat *longitude* dan *latitude*. Gambar 1. menunjukkan navigasi *waypoint*.



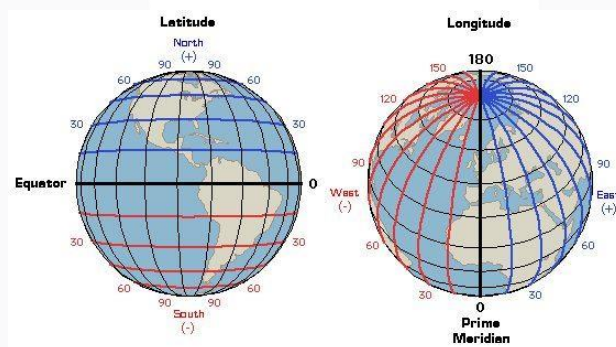
Gambar 1. Sistem Navigasi *Waypoint*

2.2. Titik Koordinat

Latitude adalah garis yang membentang secara mendatar atau horizontal di permukaan bumi. Titik 0° adalah bidang ekuator, tanda positif (+) akan menunjukkan ke arah atas bagian bumi (kutub utara), sedangkan untuk tanda negatif (-) menunjukkan ke arah bawah bagian bumi (kutub selatan). Maka dari itu titik koordinat yang digunakan adalah 0° sampai 90° untuk ke arah kutub utara dan 0° sampai -90° untuk ke kutub selatan.

Longitude adalah garis yang membentang secara vertikal di permukaan bumi. Titik diawali dari 0° sampai 180° dan 0° sampai -180° . *Longitude* ini akan menentukan lokasi di wilayah barat atau timur dari garis utara selatan yang sering disebut garis meridian.

Nantinya kedua koordinat ini akan digunakan sebagai masukan robot sebagai titik acuan. Titik acuan tersebut nantinya akan dikonversikan menjadi jarak dan sudut antara dua titik tersebut. Garis *latitude* dan garis *longitude* dapat dilihat seperti di Gambar 2.



Gambar 2. *Latitude* dan *Longitude*

2.3. Kendali PID

Untuk mengatur nilai masukan sistem dan nilai keluaran dari sensor, maka dibutuhkan sebuah pengendali. Kendali yang digunakan di penelitian ini akan berjenis PID. Kendali proporsional (P) akan mempengaruhi respon transien dan *error steady state*[6]. Berikut karakteristik dari pengendali P, dengan konstanta proporsional K_p :

- Jika nilai K_p semakin kecil, maka respon sistem akan semakin lambat
- Jika nilai K_p semakin besar, maka respon sistem akan semakin cepat
- Jika nilai K_p semakin kecil, maka nilai *error* yang dihasilkan akan semakin kecil
- Jika nilai K_p semakin besar, maka nilai *error* yang dihasilkan akan semakin besar

Berikut karakteristik dari pengendali I, dengan konstanta Integral Ki:

- Jika nilai Ki semakin besar, maka nilai *error steady state* akan semakin kecil
- Jika nilai Ki semakin kecil, maka nilai *error steady state* akan semakin besar

Berikut karakteristik dari pengendali derivatif, dengan konstanta Kd:

- Besarnya nilai Kd maka akan mengurangi osilasi yang dihasilkan sistem
- Kecilnya nilai Kd maka akan meningkatkan stabilitas sistem

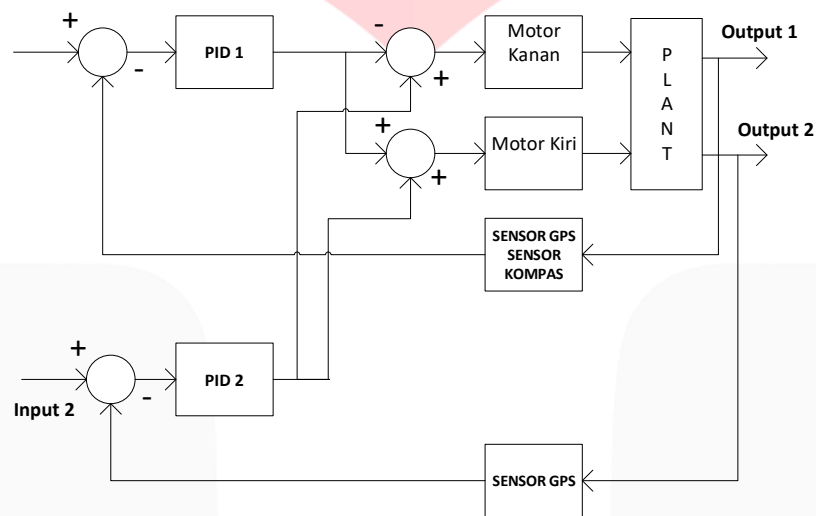
Persamaan [1] merupakan persamaan PID, dimana $e(t)$ merupakan nilai *error* terhadap waktu:

$$U(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

Untuk merancang sistem kendali PID, kebanyakan dilakukan dengan metoda coba coba atau *trial & error*. Hal ini disebabkan karena parameter K_p , K_i dan K_d tidak independent. Untuk mendapatkan aksi kendali yang baik diperlukan langkah coba-coba dengan kombinasi antara P, I dan D sampai ditemukan nilai K_p , K_i dan K_d seperti yang diinginkan[7].

3. Perancangan Sistem

Diagram blok sistem navigasi *mobile robot* ditunjukkan Gambar 3. berikut:



Gambar 3. Diagram Blok Sistem Navigasi

Keterangan:

Input 1 = Data beda sudut antara *mobile robot* dengan titik acuan

Input 2 = Data jarak antara *mobile robot* dengan titik acuan

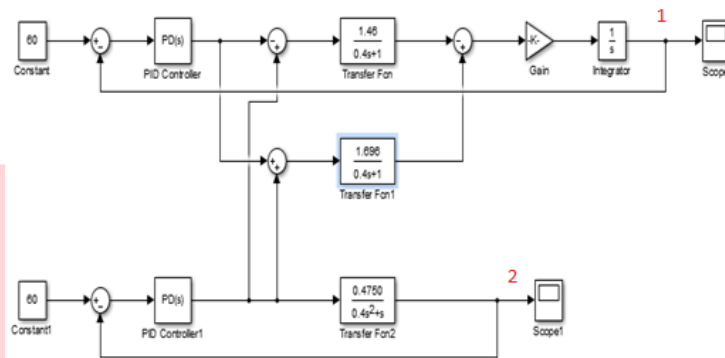
Output 1 = Sudut antara *mobile robot* dengan titik acuan

Output 2 = Jarak antara *mobile robot* dengan titik acuan

Pada diagram blok tersebut terdapat dua input yaitu data jarak antara *mobile robot* dengan titik acuan dan data beda sudut antara *mobile robot* dengan titik acuan. Data tersebut akan didapatkan dari nilai masukan (*longitude* dan *latitude*) yang kita berikan ke sistem dan diolah menjadi *error* jarak dan *error* sudut. Karena *input* berjumlah dua, maka kendali yang digunakan berjumlah dua. PID1 akan digunakan sebagai pengendali jarak antara *mobile robot* dengan titik acuan, sedangkan PID2 akan digunakan sebagai pengendali sudut antara *mobile robot* dengan titik acuan. Keluaran dari PID tersebut akan berupa PWM yang digunakan untuk menggerakkan motor dan menggerakkan *plant*. *Output* dari sistem ini adalah posisi aktual *mobile robot*.

3.1. Desain Strategi Simulasi Sistem Kendali

Dalam mencari nilai parameter K_p , K_i dan K_d dapat dilakukan dengan dua cara yaitu *trial and error* atau dengan cara simulasi. Pada penelitian ini akan menggunakan metode simulasi terlebih dahulu guna mendapatkan nilai K_p , K_i dan K_d . Berikut Gambar. 4 merupakan desain strategi simulasi sistem kendali.



Gambar 4. Desain Strategi Sistem Kendali

Gambar 4. merupakan strategi simulasi sistem kendali yang dirancang, agar didapatkan *output* berupa jarak dan sudut dari *mobile robot* terhadap titik acuan. Angka 1 pada Gambar. 4 merupakan kendali sudut dan angka 2 pada Gambar. 4 merupakan kendali jarak. Dari hasil strategi simulasi sistem kendali, maka kendali yang digunakan pada Tugas Akhir ini akan berjenis PD. Hal ini disebabkan, dari strategi simulasi yang sudah dibuat memiliki integrator, yang mana integrator adalah karakteristik dari kendali I. Dan dari hasil strategi simulasi sistem kendali yang dirancang, didapatkan nilai $K_p=7,298$ dan $K_d=1,507$.

3.2. Perancangan Perangkat Keras

Mobile robot akan dirancang dengan bahan akrilik berukuran 21 cm X 16 cm. Berikut Gambar 6. merupakan tampak depan dari *mobile robot* yang telah dirancang.



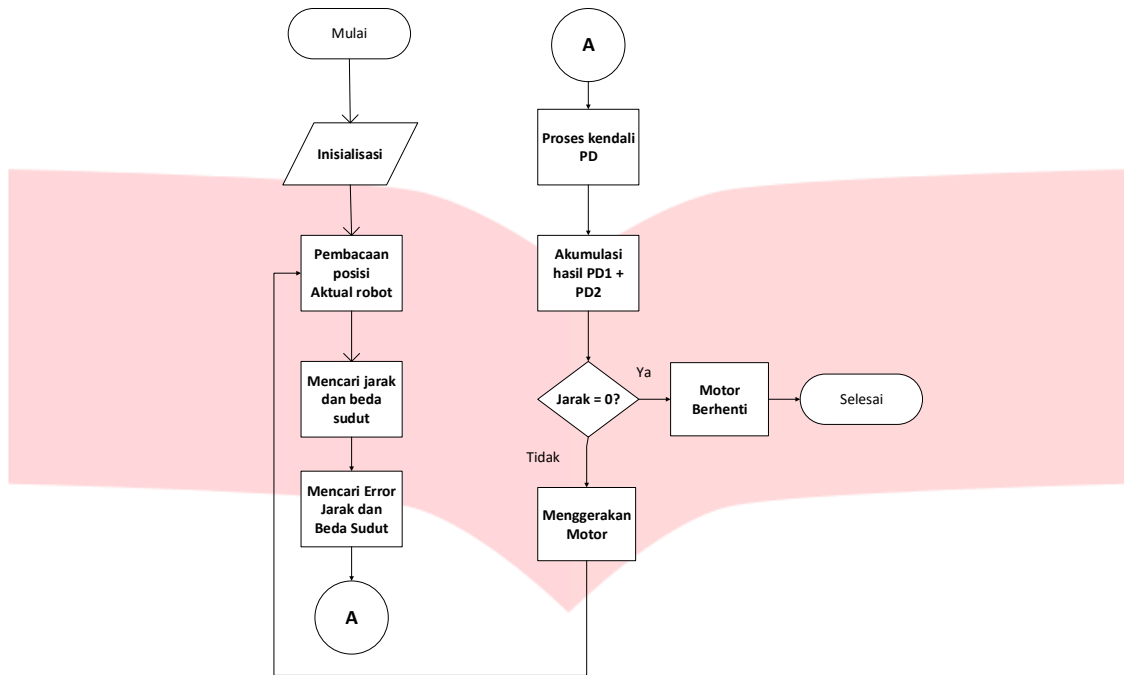
Gambar 5. Tampak Depan *Mobile Robot*

Berikut komponen yang digunakan untuk merancang *mobile robot* :

- Mikrokontroler : Arduino Uno
- Sensor GPS : U-blox Neo 6m
- Sensor Kompas : HMC5883L
- Driver Motor : L298N
- Catu daya : Baterai 12 V, 1000 mA

Merujuk pada Gambar 5, terlihat pada bagian atas *mobile robot* terdapat Rctimer NEO-6M GPS Module and MAG Compass v1.2 6M-MAG. Komponen ini diletakan paling atas robot dengan tujuan menghindari komponen tersebut dengan benda magnetik. Hal ini dikarenakan di dalam Rctimer NEO-6M GPS Module and MAG Compass v1.2 6M-MAG terdapat HMC5883L (sensor kompas) yang bekerja dengan membaca medan magnet bumi[8]. Agar kerja dari sensor kompas maksimal, maka komponen tersebut dijauhkan dari benda magnetik.

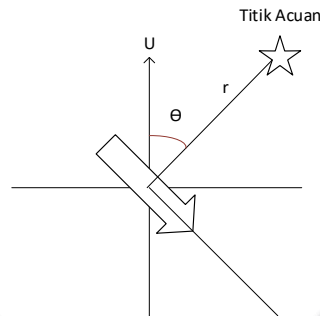
3.3. Perancangan Perangkat Lunak



Gambar 6. Algoritma Sistem

Algoritma sistem dari sistem *mobile robot* dapat dilihat pada Gambar. 6. Langkah dari setiap blok tersebut dijelaskan sebagai berikut:

- a) Inisialisasi: Ketika sistem ini dimulai, maka sistem akan membaca titik acuan yang telah di masukan oleh operator, berupa nilai *latitude* dan nilai *longitude*. Nilai tersebut akan disimpan untuk proses mencari nilai *error* jarak dan *error* sudut.
- b) Pembacaan posisi aktual *mobile robot*: Pembacaan posisi aktual dari *mobile robot* bertujuan untuk menentukan jarak dan beda sudut antara posisi aktual dengan titik acuan. Pada proses ini akan menggunakan sensor GPS untuk pembacaan posisi aktual *mobile robot* dan sensor kompas digital untuk menentukan arah orientasi robot.
- c) Mencari jarak dan beda sudut: Hasil pembacaan posisi aktual dan nilai titik koordinat acuan yang sudah disimpan akan diolah agar menghasilkan nilai jarak antara *mobile robot* dengan titik acuan dan nilai beda sudut antara *mobile robot*-titik acuan dengan arah utara.



Gambar 7. Pembacaan Sudut GPS dan Kompas

☆ = Titik Acuan

➡ = Arah Orientasi *Mobile Robot*

Θ = Pembacaan sudut antara *mobile robot*-titik acuan dengan titik 0°

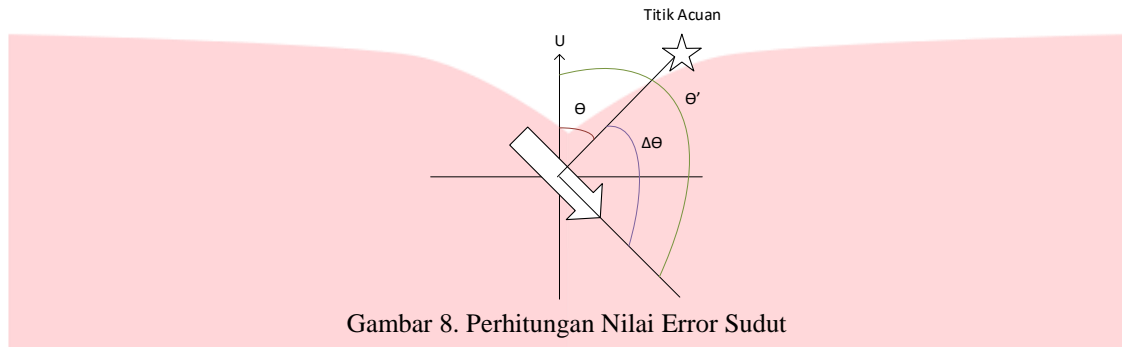
r = Jarak antara *mobile robot* dengan titik acuan

Gambar 7. menunjukkan cara sistem untuk mencari jarak antara *mobile robot* dengan titik acuan dan mencari beda sudut antara *mobile robot*-titik acuan dengan arah utara. Dalam mencari jarak, nilai *latitude* dan *longitude* dari *mobile robot* dan titik acuan akan diolah menggunakan *library TinyGPSPlus::distanceBetween*, menghasilkan jarak r . Untuk mencari beda sudut antara *mobile robot*-titik acuan terhadap arah utara dengan cara, sensor GPS akan mengambil nilai titik koordinat dari posisi *mobile robot* dan titik acuan, selanjutnya kedua titik koordinat tersebut akan diolah dengan *library TinyGPSPlus::courseTo*, menghasilkan Θ seperti di Gambar 7.

d) Mencari *error* jarak dan *error* sudut: Mendapatkan *error* jarak dapat diperoleh langsung dari langkah sebelumnya. Persamaan (12) menunjukkan *error* jarak

$$\text{error jarak} = r \quad (12)$$

Untuk mencari *error* sudut antara *mobile robot* dengan titik acuan, dapat dipahami dari Gambar 8.



Gambar 8. Perhitungan Nilai Error Sudut

$\Delta\Theta$ = *Error* sudut antara *mobile robot* dengan titik acuan

Θ' = Pembacaan sudut kompas

Untuk mendapatkan nilai *error* sudut maka dibutuhkan beberapa tahap, diantaranya:

- Pertama, sistem akan mencari arah orientasi dari *mobile robot* menggunakan sensor kompas digital. Merujuk pada Gambar 8, maka hasil dari tahap ini adalah Θ' .
- Kedua, nilai *error* sudut akan didapatkan dari pengurangan antara Θ' dengan Θ , yang akan dituliskan dengan persamaan (13)

$$\text{error sudut} = \Delta\Theta = \Theta' - \Theta \quad (13)$$

e) Proses kendali PD: *Error* jarak dan *error* sudut tersebut akan diproses dengan kendali PD. Karena terdapat dua variabel yang dikendalikan, maka PD yang digunakan pun berjumlah dua.

f) Akumulasi hasil PD1 + PD2: *Output* dari proses kendali PD akan menghasilkan PWM, masing masing PWM tersebut untuk menggerakkan dua motor (motor kanan dan motor kiri). Khusus untuk keluaran proses kendali sudut menghasilkan *output* yang bernilai negatif dan bernilai positif. Hal ini disebabkan, kendali sudut tersebut akan digunakan sistem untuk memutar arah *mobile robot*. Dan kendali jarak digunakan untuk menggerakkan *mobile robot* untuk jalan maju atau jalan mundur. *Output* PD1 digunakan sebagai pengendali jarak, menghasilkan dua *output* yang bernilai positif. Dan *output* PD2 sebagai pengendali sudut, menghasilkan dua *output* yang memiliki nilai berlawanan (negatif dan positif). Berikut persamaan (14) dan (15) memperlihatkan persamaan PWM yang digunakan untuk menggerakkan motor.

$$\text{PWM motor kanan} = \text{output PD1} + \text{output PD2} \quad (14)$$

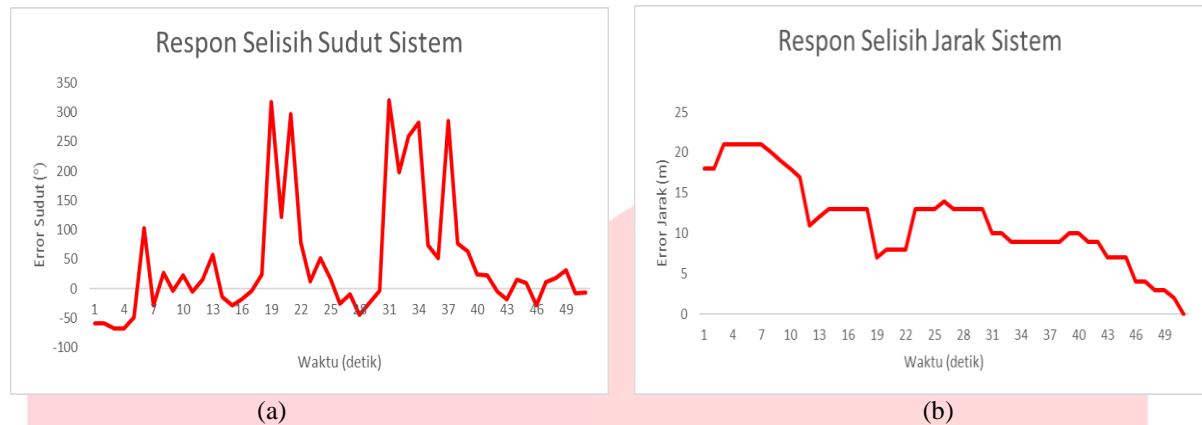
$$\text{PWM motor kiri} = \text{output PD1} - \text{output PD2} \quad (15)$$

g) Pengecekan jarak *mobile robot* dengan titik acuan (*decision*): Sebelum mengeluarkan *output* PD ke motor, sistem akan mengecek apakah masih terdapat nilai *error* jarak antara *mobile robot* dengan titik acuan apa tidak. Jika masih terdapat nilai *error* jarak maka, sistem akan menganggap *mobile robot* belum mencapai titik acuan. Jika nilai jarak = 0, maka robot sudah dianggap sampai ke titik acuan, dan motor berhenti.

h) Menggerakkan motor: Jika nilai *error* jarak masih tidak sama dengan 0, maka akumulasi PD tadi akan dikirimkan ke motor, dan motor akan berputar. Selanjutnya sistem akan kembali ke poin b hingga nilai jarak sama dengan 0.

4. Hasil Pengujian dan Analisis

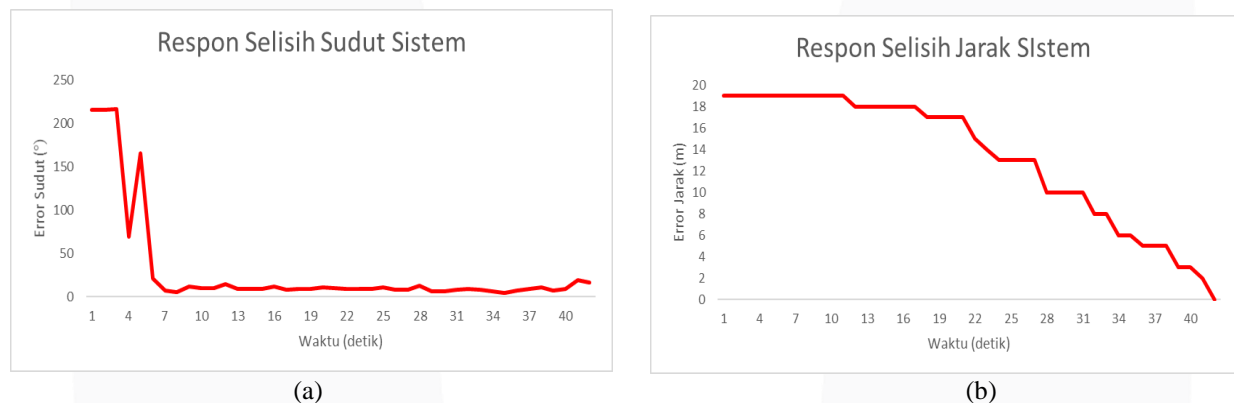
Pengujian sistem *mobile robot* yang telah dirancang dilakukan dengan cara menjalankan *mobile robot* menuju titik acuan, dan melihat respon *error* yang dihasilkan sistem. Nilai K_p dan K_d yang digunakan sesuai dengan hasil dari simulasi yaitu $K_p=7,298$ dan $K_d=1,507$. Respon yang akan dilihat berupa respon *error* jarak antara *mobile robot* dengan titik acuan, dan respon *error* sudut *mobile robot* dengan titik acuan. Berikut hasil pengujian yang dihasilkan.



Gambar 9. (a) Respon Selisih Sudut dengan $K_p=7,298$ $K_d=1,507$ (b) Respon Selisih Jarak dengan $K_p=7,298$ $K_d=1,507$

Gambar 9.(a) dan Gambar 9.(b) menunjukkan nilai selisih dari sudut dan jarak *mobile robot* ke titik acuan dengan nilai $K_p = 7.2988$ dan $K_d = 1.507$. Dilihat dari Gambar 9.(a), terdapat di beberapa titik yang menunjukkan robot berbelok arah dengan cukup signifikan. Hal itu dapat disimpulkan karena terjadi penurunan dan kenaikan grafik yang tajam pada beberapa titik. Merujuk Gambar 9.(b), *mobile robot* berjalan mendekati titik acuan, meskipun pada beberapa titik, nilai selisih jarak menunjukkan kenaikan. Sistem membaca jarak antara *mobile robot* dengan titik acuan sejauh 18 m, dan mencapai ke titik acuan 1 dalam 51 detik.

Dari pengujian tersebut, sistem sudah menunjukkan bahwa *mobile robot* dapat bergerak menuju titik acuan. Namun dalam perjalanannya menuju titik acuan, robot sering kali berbelok arah dengan sangat signifikan. Oleh karena itu, penulis akan merubah nilai K_p dan K_d yang telah dirancang. Perubahan nilai ini akan menggunakan metode *trial and error*. Dengan metode tersebut, didapatkan nilai $K_p = 0.75$ dan $K_d = 0.2$. Berikut data hasil dari pengujian tersebut.



Gambar 10. (a) Respon Selisih Sudut dengan $K_p=0,75$ $K_d=0,2$ (b) Respon Selisih Jarak dengan $K_p=0,75$ $K_d=0,2$

Gambar 10.(a) dan Gambar 10.(b) menunjukkan respon selisih sudut dan jarak dari *mobile robot* menuju titik acuan 1 dengan nilai $K_p = 0.75$ dan $K_d = 0.2$. Gambar 10.(a) menunjukkan respon yang stabil. Meskipun *mobile robot* tidak bergerak dengan lurus sempurna, tetapi tidak ada perubahan arah robot secara signifikan. Di Gambar 10.(b) pun juga menunjukkan respon yang lebih baik. Tidak ada nilai yang menunjukkan bahwa robot menjauhi titik acuan. Waktu tempuh yang dihasilkan pun juga lebih cepat, yaitu 42 detik. Hal ini dapat terjadi karena tidak adanya perubahan arah robot secara signifikan.

Dapat ditarik kesimpulan bahwa nilai $K_p=0,75$ dan $K_d=0,2$ menghasilkan respon yang lebih stabil dari pada nilai K_p dan K_d yang dihasilkan simulasi Matlab. Hal ini dapat terjadi karena banyak faktor, diantaranya gaya gesek antara lintasan dengan roda dan berat dari *mobile robot* yang tidak dimasukkan dalam strategi simulasi sistem.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Setelah membuat perancangan sistem dan melakukan pengujian serta analisis dari sistem *mobile robot* dengan menggunakan kendali PID, maka dapat ditarik kesimpulan bahwa:

1. Sistem *mobile robot* dapat bergerak menuju titik acuan dengan menggunakan kendali PD, dengan nilai $K_p=0,75$ dan $K_d=0,2$. *Mobile robot* mencapai titik acuan dengan waktu 42
2. Terdapat banyak faktor yang menyebabkan hasil simulasi Matlab tidak sesuai dengan respon yang diinginkan, seperti gaya gesek roda dengan lintasan dan massa *mobile robot* yang belum diperhitungkan

5.2. Saran

Dari hasil pengujian yang telah dilakukan, Tugas Akhir ini masih memiliki beberapa kekurangan. Tentunya dengan memperbaiki kekurangan tersebut dapat mengembangkan alat yang sudah dibuat. Berikut beberapa saran penulis, yang dapat dikembangkan di penelitian berikutnya:

1. Sistem *mobile robot* ditambahkan fitur *obstacle avoiding robot*. Sehingga selain robot dapat berpindah dari titik awal ke titik acuan, robot juga dapat menghindari objek yang terdeteksi sistem
2. Sistem *mobile robot* ditambahkan sebuah sensor lokal untuk memperkecil nilai error antara posisi akhir *mobile robot* dengan titik acuan
3. Disarankan menggunakan sensor GPS yang memiliki spesifikasi yang lebih baik, karena hal ini penting dalam mempengaruhi tingkat akurasi dan tingkat presisi sistem
4. Pembuatan dimensi *mobile robot* lebih diperbesar seperti AGV (*automatic guided vehicle*), agar robot dapat membawa beban yang besar

Daftar Pustaka

- [1] D. Liani, A. Silvia and L. , "Sistem Navigasi pada Mobile Robot dengan Global Positioning System (GPS)," *ANNUAL RESEARCH SEMINAR 2016*, vol. 2, no. 1, p. 1, 2016.
- [2] Z. and U. Sutisna, "Perancangan Sistem Kontrol Wireless pada Mobile Robot Manipulator Berbasis Mikrokontroler ATmega," *JNTETI*, vol. 3, no. 1, p. 1, 2014.
- [3] H. Wu, W. Su and Z. Liu, "PID controllers: design and tuning methods," *IEEE*, 2014.
- [4] A. S. Taufik, "Sistem Navigasi Waypoint pada Autonomous Mobile Robot," p. 1.
- [5] I. Marito, *SISTEM NAVIGASI HELIKOPTER BERDASARKAN DATA POSISI SECARA TELEMETRI*, Depok, 2008.
- [6] F. Vahid and T. Givargis, *EMBEDDED SYSTEM DESIGN*, United States Of America: John Wiley & Sons, INC, 2002.
- [7] M. Ali, "PEMBELAJARAN PERANCANGAN SISTEM KONTROL PID DENGAN SOFTWARE MATLAB," *Jurnal Edukasi@Elektro*, vol. 1, no. 1, 2004.
- [8] Febriyansyah, *APLIKASI SENSOR KOMPAS MAGNOMETER 3 AXIS HMC5883L PADA PROTOTYPE ROBOT BOAT PENGUMPUL SAMPAH BERBASIS MIKROKONTROLER*, PALEMBANG, 2014.