

Sistem Navigasi Robot Berkaki Menggunakan Sensor Lidar Dengan Metode PID

Alfian Nurul¹, Totok Winarno², Achmad Komarudin³

[Submission: 07-05-2021, Accepted: 30-05-2021]

Abstract – The legged robot is a robot that functions to explore space by space using a servo-like motion that resembles a leg. In order to explore and move as desired without crashing, a robotic navigation system is needed. In this study using a lidar sensor with the PID method. The PID method is used because it is efficient for straightening the movement of the robot while walking along the walls of the race area. From the experimental analysis that has been done, it is known that the use of the PID controller is very influential on robot navigation. The use of the PID algorithm makes the legged robot more stable in navigating from room to room. The robot will stabilize the navigation motion according to the desired setpoint. So that the robot does not touch or hit the wall on the track, a lidar sensor is needed. The lidar sensor was chosen because this sensor can make the controller work easier, besides the blind spot of the robot gets smaller if you use one sensor to access the distance within 360° so that the robot will be more reliable in tracking the track. After testing, the PID value at a distance of 18 cm for the left alignment is $K_p = 0.3$, $K_i = 0.2$, and $K_d = 4.5$ and for the right average $K_p = 0.2$, $K_i = 0.3$ and $K_d = 7.5$ by doing 3 times the road can move without hitting the wall with an error of 10 - 20% due to reduced resources and slippage.

Intisari — Robot berkaki merupakan robot yang berfungsi menjelajahi ruang demi ruang dengan menggunakan penggerak berupa servo yang menyerupai sebuah kaki. Agar dapat menjelajahi dan bergerak sesuai dengan keinginan tanpa menabrak, maka dibutuhkan sistem navigasi robot. Pada penelitian ini menggunakan sensor lidar dengan metode PID. Metode PID digunakan karena efisien untuk meluruskan pergerakan robot saat menyusuri dinding area perlombaan. Dari analisa percobaan yang sudah dilakukan diketahui bahwa penggunaan kontroler PID sangat berpengaruh terhadap navigasi robot. Penggunaan algoritma PID menjadikan robot berkaki lebih stabil melakukan navigasi dari ruang ke ruang. Robot akan menstabilkan gerak navigasi sesuai dengan setpoint yang diinginkan. Agar robot tidak menyentuh atau menabrak dinding pada lintasan maka diperlukan sensor lidar. Sensor lidar dipilih karena sensor ini dapat membuat kerja kontroler lebih mudah, selain itu titik buta dari robot semakin kecil jika memakai satu buah sensor untuk mengakses jarak dalam 360° sehingga robot akan lebih handal dalam menelusuri lintasan. Setelah dilakukan pengujian dihasilkan nilai PID pada jarak 18 cm untuk rata kiri adalah $K_p = 0.3$, $K_i = 0.2$, dan $K_d = 4.5$ dan untuk rata kanan $K_p = 0.2$, $K_i = 0.3$ dan $K_d = 7.5$ dengan melakukan 3 kali jalan dapat bergerak tanpa menabrak dinding dengan error 10 - 20% dikarenakan sumber dari robot berkurang dan selip.

Kata Kunci — Robot Berkaki, PID, Lidar, Wall Following

I. PENDAHULUAN

Salah satu dari ilmu pengetahuan dan teknologi yang

digunakan untuk mengembangkan ide yaitu dari ilmu elektronika, khususnya bidang ilmu robotika. Robot merupakan alat yang digunakan untuk membantu dan meringankan tugas dari manusia sehingga lebih efektif dan efisien dalam menyelesaikan suatu kegiatan. Robot dapat dibedakan berdasarkan alat geraknya dibagi menjadi dua, yaitu robot berkaki (lengan) dan robot beroda.

Di Indonesia, dunia robotika telah berkembang pesat dengan adanya salah satu wadah pengembangan teknologi robotika di bidang pendidikan adalah Kontes Robot Indonesia (KRI) yang diadakan oleh Direktorat Jenderal Pendidikan Tinggi (Dirjen DIKTI), yang terdiri dari 5 kategori, yaitu Kontes Robot ABU Indonesia (KRAI), Kontes Robot Pemadam Api Indonesia (KRPAI), Kontes Robot Sepak Bola Indonesia (KRSBI), Kontes Robot Seni Tari Indonesia (KRSTI), dan Kontes Robot Tematik Indonesia (KRTMI) yang diselenggarakan setiap satu tahun sekali.

Pada Kontes Robot Pemadam Api Indonesia (KRPAI) robot harus menggunakan kaki untuk bergerak (*actuator* tidak diperbolehkan berputar 360 derajat). Robot harus mampu beradaptasi dan melaksanakan tugasnya untuk menyusuri arena yang berupa beberapa ruangan dan mendeteksi api dalam satu ruangan untuk dipadamkan.

Penggunaan sensor lidar (LDS-01) memiliki beberapa manfaat yaitu dapat membuat kerja kontroler lebih mudah karena data sudah siap diolah dan telah diproses oleh *miniPC* sehingga tidak memberatkan kontroler untuk mengaksesnya. Keuntungan yang lain adalah titik buta dari robot semakin kecil dengan memakai satu buah sensor untuk mengakses jarak dalam 360°. Hal tersebut diharapkan mampu mengatasi masalah yang lebih handal agar robot tidak menyentuh dinding dan lebih aman.

Dalam Kontes Robot Pemadam Api Indonesia (KRPAI) devisi berkaki memiliki ketentuan bahwa robot harus bergerak secara otomatis dalam menelusuri dinding, sehingga dibutuhkan suatu kontrol. Kontrol yang sangat efisien bagi sistem ini adalah kontrol PID. Hal tersebut sesuai dengan penelitian yang sudah dilakukan, terbukti bahwa kontroler PID dapat digunakan untuk meluruskan pergerakan robot saat menyusuri dinding arena perlombaan. Dengan menggunakan metode *hand tuning* dapat menentukan hasil

p-ISSN: 2356-0533; e-ISSN: 2355-9195

¹Mahasiswa, Program Studi Teknik Elektronika Jurusan Teknik Elektro Politeknik Negeri Malang, Jl. Soekarno Hatta No. 9 (telp: 0341-404424 ; e-mail: alfiannurul02@gmail.com)

^{2,3}Dosen, Program Studi Teknik Elektronika Jurusan Teknik Elektro Politeknik Negeri Malang, Jl. Soekarno Hatta No. 9 (telp: 0341-404424 ; e-mail: totok.winarno@polinema.ac.id, e-mail: achmad.komarudin@polinema.ac.id)



parameter kontroler PID, sedangkan pengontrolnya dilakukan dengan penentuan masukan parameter berupa jarak yang diinginkan terhadap dinding (*setpoint*). Selanjutnya data yang berupa jarak tersebut diolah menggunakan kontroler yang menghasilkan sinyal untuk mengontrol *plan* yang dapat menentukan keluaran posisi jarak yang terukur dari robot.[1]

Dalam metode PID nilai *error* yang akan didapat dari perbedaan *setpoint* dengan nilai *actual* dari sensor akan memberikan suatu *feedback* yang mampu untuk diolah oleh *microcontroller* agar *actuator* berjalan sesuai dengan kehendaknya.

II. KAJIAN PUSTAKA

A. PID (Proportional Integral Derivative) Controller

Pada suatu sistem kontrol dibutuhkan keberadaan kontroler yang memiliki peran besar terhadap perilaku dari sistem. Hal ini disebabkan oleh tidak dapat diubahnya suatu komponen penyusun sistem, sehingga karakteristik dari *plant* yang akan dikontrol tidak dapat diubah. Maka perubahan perilaku sistem dapat dilakukan melalui kontroler.[2]

Kontroler *Proportional* memiliki keluaran yang sebanding dengan besarnya sinyal kesalahan (selisih dari nilai yang diinginkan dengan nilai *actual*). Secara sederhana dapat dikatakan bahwa keluaran kontrol proporsional merupakan perkalian dari konstanta proporsional dengan masukannya.[3]

Kontroler *Integral* berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan mantap nol. Jika sebuah *plant* tidak memiliki unsur integrator ($1/s$), kontroler proporsional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan mantapnya nol.[4]

Kontroler *Diferensial* memiliki keluaran bersifat seperti suatu operasi derivatif. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat.[5]

B. Laser Odometry

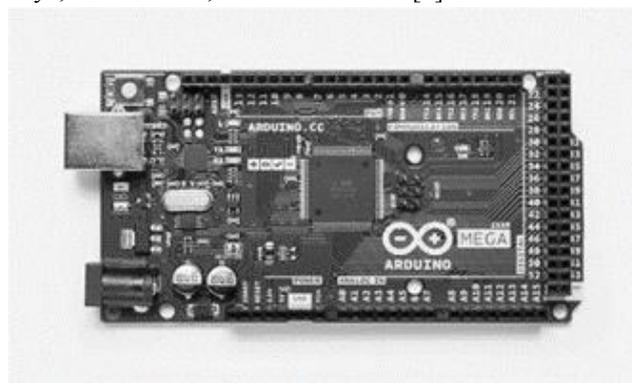
Odometry adalah sebuah pengukuran yang menggunakan data sensor untuk mengetahui posisi robot terhadap sebuah lingkungan (*environment*) yang mana memiliki acuan yang tetap (statik) terhadap lingkungannya. Sensor yang dapat digunakan sebagai *odometry* adalah *encoder*, laser, kamera, *gyro*, *accelero*, dan yang lainnya. [6] Salah dari *odometry* sensor yaitu laser *odometry*. Laser *odometry* adalah membandingkan nilai jarak yang terbaca oleh sensor dengan nilai jarak yang terbaca sebelumnya dalam sampling waktu tertentu.



Gambar 1. Odometry Laser

C. Arduino Mega 2560

Arduino Mega 2560 merupakan sebuah papan atau rangkaian *microcontroller* berbasis Atmega 2560. Arduino ini, memiliki 54 digital pin *input/output* (14 dapat digunakan sebagai *output* PWM), 16 analog *input*, 4 UART (*hardware port serial*), 16 MHz *osilator* kristal, koneksi USB, jack catu daya, header ICSP, dan tombol reset. [7]



Gambar 2. Arduino Mega 2560

D. Odroid XU4

Odroid XU4 adalah salah satu perangkat komputasi papan tunggal (*mini PC*) terbaru yang sangat hemat energi dan kuat, memiliki bentuk yang kecil jika dibandingkan dengan perangkat tertanam berbasis ARM yang lain. Odroid juga mendukung sistem operasi *open source* dan menjalankan berbagai versi dari linux termasuk Ubuntu dan berbagai versi Android termasuk Lollipop. Tidak seperti Raspberry, Odroid tidak memiliki WiFi dan Bluetooth pada *board*-nya. Maka kita perlu USB WiFi *adapter* jika ingin menggunakan internet.[8]





Gambar 3. Odroid XU4

E. Sensor Lidar (LDS-01)

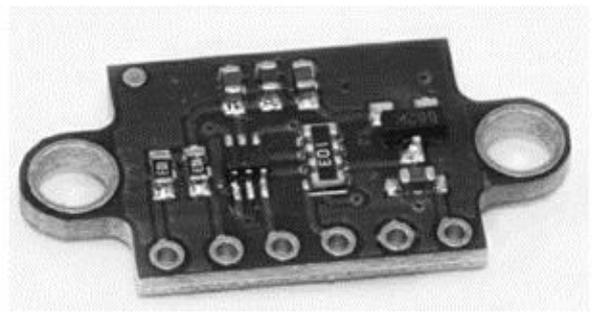
LDS-01 adalah sensor laser 2D yang mampu untuk berputar dan mendeteksi jarak dengan sudut 360° dengan kemampuan mendeteksi sejauh 3 meter dan mampu untuk menyimpan data yang akan digunakan untuk SLAM (*Simultaneous Localization and Mapping*). [9]



Gambar 4. Sensor Lidar (LDS-01)

F. *Sensor Time Of Flight* (TOF) VI53L0X

Sensor Time Of Flight adalah sebuah sensor jarak yang melakukan pengukuran dengan cara menghitung waktu yang dibutuhkan cahaya inframerah yang mengenai suatu objek untuk mengetahui posisi jaraknya. Sensor ini memiliki kecepatan melebihi sensor ultrasonic karena menggunakan cahaya inframerah. Sensor VI53L0X memiliki resolusi ketelitian 1 mm serta memiliki range atau jarak maksimal 2 meter. [10]



Gambar 5. *Sensor Time Of Flight*

G. Servo AX-12A

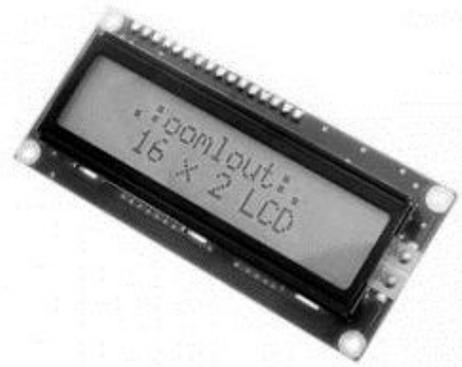
Servo Dynamixel AX-12 (*Servo Robotic Bioloid series*) adalah *aktuator* berbasis motor servo yang sudah terdapat kontroler di dalamnya sehingga dikatakan sebagai *smart servo*. Cara mengakses servo ini dengan menggunakan komunikasi *serial half duplex* dengan kecepatan mencapai 1Mbps. [11]



Gambar 6. Servo AX-12A

H. *Liquid Crystal Display (LCD) 16 x 2*

LCD adalah suatu jenis perangkat elektronik yang dapat menampilkan kode ASCII (*American Standart Code for Information Interchange*) yang dapat membantu pengguna sebagai media antar muka antara pengguna dan mesin untuk menyampaikan data yang sedang diolah, menu dalam sistem, serta mengatur parameter dari sistem saat sistem sedang bekerja. [12]



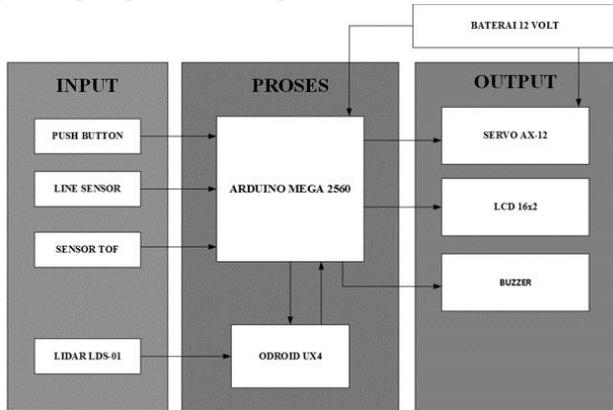
Gambar 7. LCD 16x2



III. METODE PENELITIAN

A. Diagram Blok Sistem

Gambar 8 merupakan diagram blok sistem dari penelitian ini. Diagram blok sistem ini mempunyai 3 blok, yaitu *input*, proses dan *output*.

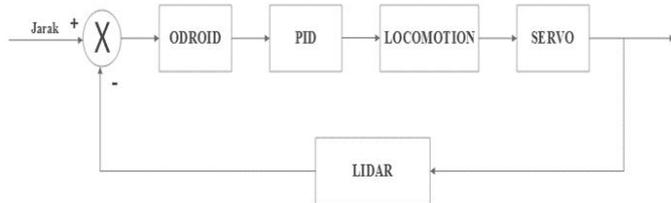


Gambar 8. Diagram Blok Sistem

Dari diagram blok tersebut, prinsip kerja robot ini adalah menggunakan sensor jarak sebagai input PID yang akan diproses oleh kontroler untuk menjadi masukan bagi *actuator* yang berupa servo, sehingga robot dapat bergerak mengikuti dinding.

B. Diagram Blok Kontrol

Gambar 9 merupakan diagram blok kontrol yang digunakan untuk sistem ini



Gambar 9. Diagram Blok Kontrol

C. Prinsip Kerja

Prinsip kerja dari navigasi robot menggunakan metode *Wall Following* saat robot pertama kali dijalankan akan mendeteksi berada di ruang mana terlebih dahulu. Cara mengetahui keberadaan robot adalah dengan cara mendeteksi apakah robot berada lurus dengan dinding atau tidak. Setelah robot lurus dengan dinding maka dilakukan pengukuran menggunakan sensor jarak untuk mendeteksi robot berada didalam ruangan yang mana.

Saat robot telah mengetahui keberadaan posisinya maka robot tersebut akan berjalan sesuai dengan program yang telah diberikan di dalamnya. Robot akan memilih untuk menggunakan rata kanan atau rata kiri sesuai dengan titik acuan yang telah dimasukkan di dalam program. Titik acuan

tersebut berupa pengukuran terhadap lorong dan ruangan. Robot akan menyusuri tiap ruang untuk mencari ruangan yang diinginkan. Bila robot sudah menemukan ruangan tujuan maka robot akan berhenti.

D. Perancangan Mekanik

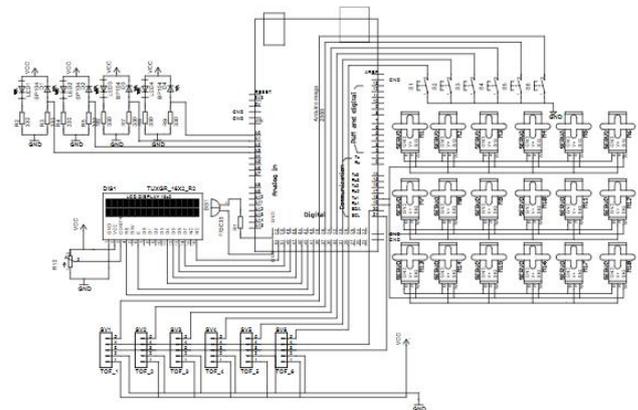
Gambar 10 merupakan desain mekanik robot dari samping atas yang akan digunakan.



Gambar 10. Desain Mekanik

Pada perancangan mekanik, robot akan menggunakan 6 kaki yang terdiri dari 3 servo dalam setiap kakinya. Pada bagian atas robot terdapat sensor Lidar, agar dalam proses mengukur jarak antara robot dengan dinding tidak terhalangi oleh *body* dari robot. Dan terdapat LCD untuk melakukan pengaturan dan melihat parameter saat robot berjalan.

E. Perancangan Rangkaian Shield Arduino Mega



Gambar 11. Perancangan Rangkaian Shield Arduino Mega

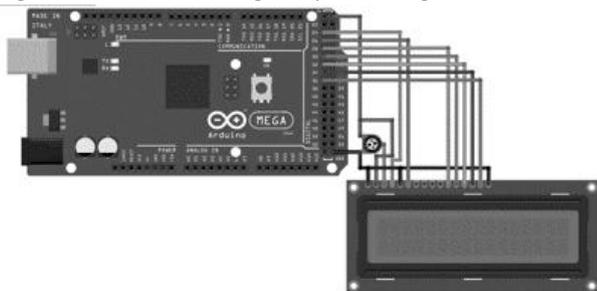
Rangkaian Arduino Mega digunakan sebagai master dalam robot. Arduino mega tersambung dengan LCD, *button*, Sensor TOF, Sensor Garis, Servo, dan Buzzer. Maka untuk mempermudah menyambungkan dan meminimalisasi kesalahan menyambungkan berbagai komponen ditambahkan *shield Arduino Mega*. Dengan membuat *shield* sendiri dapat memodifikasi sesuai dengan kebutuhan kita sendiri agar lebih



efisien dan hemat dalam penggunaan tempat karena dimensi robot yang kecil serta mengurangi masalah dalam pengkabelan tiap komponen dalam robot.

F. Perancangan Rangkaian LCD 16x2

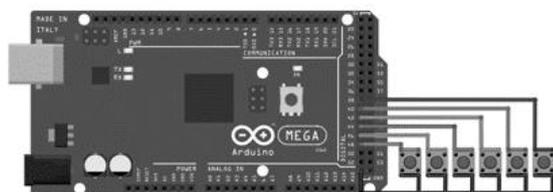
Untuk mempermudah dalam melihat parameter dan mengganti nilai variabel dibutuhkan *display* yang gampang diakses. Maka dibutuhkan LCD ukuran 16x2. Penggunaan LCD ini tidak memakan tempat dan pas digunakan di robot. Di dalam sini akan menampilkan posisi awal robot dan posisi tujuan robot serta menampilkan nilai pid yang digunakan. Rangkaian LCD sesuai dengan *layout* sebagai berikut:



Gambar 12. Perancangan Rangkaian LCD 16x2

G. Perancangan Rangkaian Push Button

Untuk memudahkan mengganti menu tampilan pada LCD, serta mengganti nilai parameter maka diperlukan sebuah *input* yang dapat mempermudah kita dalam menggunakannya. *Input*-an yang paling sederhana dan gampang adalah *push button*. Berikut skema *push button* dengan Arduino Mega:

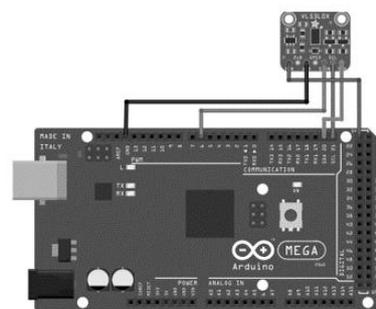


Gambar 13. Perancangan Rangkaian Push Button

Pada penggunaan robot nanti digunakan beberapa tombol navigasi untuk memindahkan kursor dan masuk kedalam menu robot. Tombol yang digunakan antara lain adalah *enter*, *back*, atas, bawah, kiri, dan kanan.

H. Perancangan Rangkaian Sensor TOF

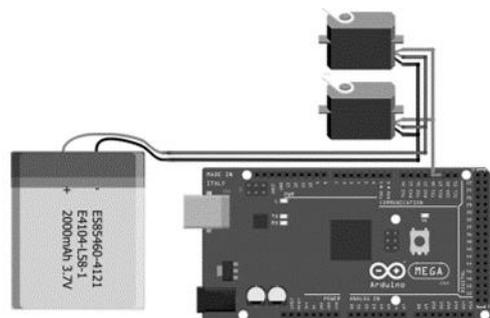
Sensor ini menggunakan inframerah yang memiliki kecepatan lebih cepat dari pada menggunakan sensor ultrasonik. Untuk mengakses sensor ini menggunakan komunikasi I2C. Setiap sensor akan memiliki *address* yang berbeda – beda sesuai dengan sambungan *pin xshunt* ke dalam pin Arduino. Berikut sambungan antara Arduino Mega dengan Sensor TOF :



Gambar 14. Perancangan Rangkaian Sensor TOF.

I. Perancangan Rangkaian Servo AX-12

Servo ini menggunakan tegangan kerja 12 volt. Dengan menggunakan komunikasi serial maka menghemat penggunaan pin yang dibutuhkan. Sambungan untuk servo ini langsung menyambungkan pin RX1 dan TX1. Untuk pembagian waktu kapan servo *transmit* atau *receive* sudah diatur oleh program di dalam *library* yang ada di Arduino. Berikut merupakan skematik dari sambungan antara Arduino dan servo AX-12 :



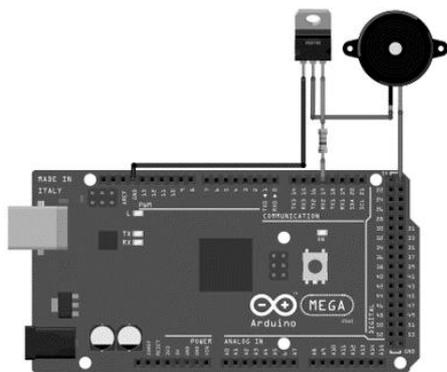
Gambar 15. Perancangan Rangkaian Sensor TOF

Untuk menggerakkan tiap servo dengan memberikan atau mengatur *baudrate* atau kecepatan *transfer* data dan mengatur alamat tiap servo terlebih dahulu. Setiap servo tidak boleh memiliki alamat yang sama agar robot dapat berjalan dengan benar.

J. Perancangan Rangkaian Buzzer

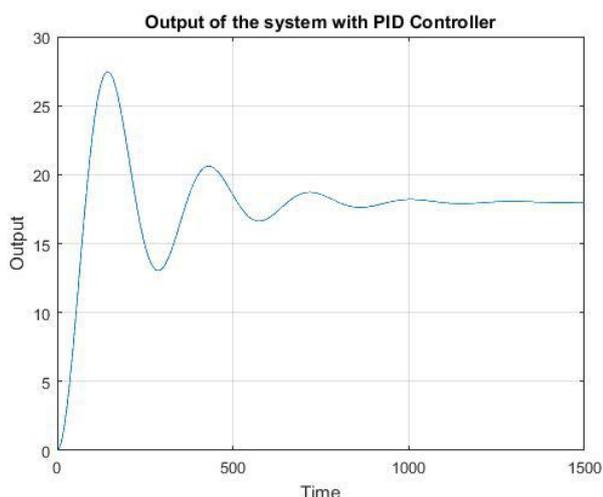
Sebagai indikator apabila robot sudah sampai di tempat tujuan maka dibutuhkan buzzer sebagai pertanda. Agar mempermudah memastikan apakah robot benar – benar sudah selesai menjalankan perintah atau masih belum selesai menjalankan perintah yang telah ditujukan. Berikut merupakan rangkaian yang digunakan:





Gambar 16. Perancangan Rangkaian Buzzer

K. Perancangan Kontrol PID



Gambar 17. Hasil Perancangan Kontrol PID

Pada perancangan PID, menggunakan simulasi Matlab dengan menggunakan *transfer function* sebagai berikut :

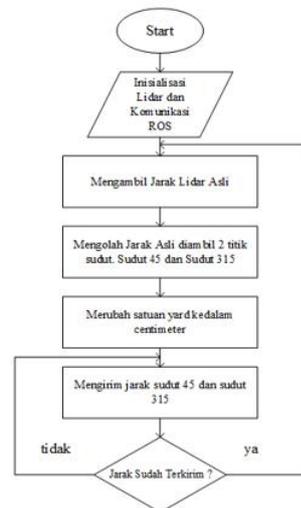
$$\frac{K_d s^2 + K_p s + K_i}{s^3 + K_d s^2 + K_p s + K_i} \quad (1)$$

Transfer function tersebut akan di simulasikan menggunakan m file pada Matlab. Dimana untuk nilai $K_p = 0.8$; $K_i = 0.1$; $K_d = 1.2$; $num = [K_p \ K_i \ K_d]$; $den = [1 \ K_p \ K_i \ K_d]$; $t = 0:100:2000$; $step(num,den)$; disini menggunakan $Ref = 18$ dimana akan menjadi acuan *setpoint* pada robot nantinya.

Pada gambar 17 merupakan output dari *transfer function* yang ada pada persamaan (1). Pada gambar memperlihatkan simulasi keluaran sinyal sampai 1500ms (1,5 detik). Terlihat bahwa sinyal awal memiliki overshoot dan mendekati setpoint sehingga sinyal tersebut dapat digunakan sebagai acuan nilai awal yang dapat dimasukkan kedalam kontrol PID robot.

Sehingga dapat memudahkan kita mencari nilai yang cocok untuk robot. Akan tetapi pada waktu pengujian terkadang nilai yang dibutuhkan oleh robot tidak akan sama persis dengan simulasi dan juga tidak terlalu jauh dari nilai simulasi yang telah dibuat menggunakan matlab. Oleh karena itu perlu dilakukan pengujian terlebih dahulu.

L. Perancangan Software



Gambar 18. Perancangan Software

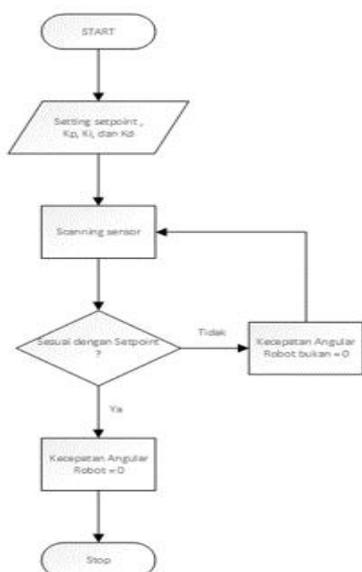
Pada gambar 18 merupakan *flowchart* perancangan *software* yang berada pada odroid. Saat robot dinyalakan Odroid akan mengambil data dari lidar secara berkala. Data dari lidar tersebut akan diolah untuk diambil pada jarak sudut 45° dan 315° . Data tersebut masih dalam satuan *yard* dan akan di ubah menjadi satuan sentimeter.

Setelah data lidar diubah dalam satuan sentimeter maka Odroid akan mengirim data tersebut ke Arduino Mega menggunakan Komunikasi ROS Serial. Pengiriman tersebut setiap 100 ms. Apabila data jarak sudah dikirim ke Arduino maka Arduino akan memberikan *feedback* ke Odroid. Apabila arduino belum mengirimkan *feedback* maka Odroid tidak akan mengirim data terlebih dahulu agar tidak ada penumpukan data (*buffer*).

M. Perancangan Algoritma Navigasi Robot

Untuk dapat memberikan perintah kepada robot agar dapat bergerak menelusuri lorong dan ruang, mendeteksi lokasi atau tiap ruang yang berbeda maka diperlukan sebuah algoritma pemrograman untuk dapat menyelesaikan pekerjaan tersebut.





Gambar 19. Perancangan Rangkaian *Software*

Pada gambar 19 merupakan *flowchart* perancangan *software* yang berada pada Arduino Mega. Saat Arduino pertama kali dijalankan maka akan mendeklarasi variabel serta mengatur nilai Kp, Ki, Kd, dan *setpoint*. Setelah itu Arduino akan menunggu data *scanning* jarak yang dikirimkan oleh Odroid. Apabila data sudah terkirim maka akan menjadi nilai *present value* yang akan diselisihkan oleh *setpoint*.

Jika nilai selisih tersebut 0 (nilai *error*) maka *output* dari kontrol PID juga 0 sehingga robot akan berjalan lurus tanpa ada kecepatan *angular*. Namun jika nilai selisih tersebut bukan sama dengan 0 maka akan diproses oleh kontrol PID sehingga robot akan mempunyai kecepatan *angular* (belok) sesuai dengan nilai *output* dari controller PID.

IV. ANALISA DAN PEMBAHASAN

A. Pengujian dan Analisa Komunikasi ROS Serial.

Komunikasi ROS serial dilakukan antara Arduino Mega dengan Odroid XU4. Pengujian dilakukan untuk mengetahui apakah data yang dikirimkan dan yang diterima sama antara Arduino Mega dengan Odroid XU4.

TABEL 1.

HASIL PENGIRIMAN DATA KOMUNIKASI ROS SERIAL

Data Jarak Pada Odroid UX4	Data Jarak Pada Arduino Mega	Error(%)
10	10	0
15	15	0
20	20	0

25	25	0
30	30	0
35	35	35
40	40	40
45	45	45
50	50	50
Rata – Rata Error		0

Pada pengujian tersebut Arduino Mega dengan Odroid XUA4 dapat berkomunikasi dan data dapat terkirim sesuai keinginan tanpa ada data yang berubah nilainya. Maka Odroid UX4 dapat mengirim data lidar yang telah diolah dan Arduino Mega dapat menerima sebagai input jarak yang akan diproses menggunakan kontrol PID.

Pada pengujian berikut merupakan pengujian selang waktu pengiriman data pada Odroid UX4 dan Arduino Mega pengiriman dilakukan setiap 100 ms. Berikut hasil pengujian selang waktu pengiriman data.

TABEL 2.

HASIL PENGUJIAN SELANG WAKTU PENGIRIMAN DATA

Waktu pada Odroid UX4 (ms)	Waktu pada Arduino Mega (ms)	Selisih (ms)	Error(%)
100	100	0	0
100	100	0	0
100	104	4	4
100	105	5	5
100	120	20	20
100	115	15	15
100	100	0	0
100	100	0	0
100	103	3	3
Rata – Rata Error			5.22

Pada pengujian tersebut Arduino Mega dengan Odroid XUA4 dapat saling berkomunikasi akan tetapi ada *delay* waktu yang diperlukan untuk mengirim data. Pada pengiriman data terdapat error waktu sebanyak 5.22%. waktu tunda tersebut masih tergolong cepat dalam (ms).

B. Pengujian dan Analisa Sensor Lidar



Pengujian dilakukan pada 2 buah titik sudut yaitu 0 derajat (depan robot) dan 270 derajat (kanan robot). Dalam Pengujian ini sensor diatur agar mengirim jarak dalam satuan cm.

TABEL 3.
HASIL PENGUKURAN SENSOR LIDAR DEPAN

Jarak Aktual (cm)	Jarak terukur (cm)	Error(%)
10	10	0
15	15	0
20	20	0
25	24	4
30	31	3.33
35	36	2.86
40	40,5	1.25
45	46	2.22
50	51	2
Rata – Rata Error		1.74

TABEL 4.
HASIL PENGUKURAN SENSOR LIDAR KANAN

Jarak Aktual (cm)	Jarak terukur (cm)	Error(%)
10	10	0
15	15	0
20	20	0
25	25	0
30	30	0
35	36	2.86
40	40,5	1.25
45	46	2.22
50	51	2
Rata – Rata Error		0.925

Berdasarkan hasil pengujian sensor lidar dengan menggunakan sudut 0 derajat (depan robot) pada tabel 1 dan dengan sudut 270 derajat (kanan robot) pada tabel 2, maka dapat diketahui bahwa sensor lidar memiliki kepresisian yang tinggi jika jarak yang diukur dekat. Semakin jauh jarak yang diukur, kepresisiannya akan semakin berkurang dan

akan terdapat error.

Dalam pengujian yang dilakukan pada sensor depan dan sensor kanan dengan pengukuran pada jarak 10 cm hingga 50 cm dengan interval 5 cm didapatkan hasil rata-rata error sebesar 1,74% dan 0.925%. Dari hal tersebut, diketahui bahwa sensor lidar pada kanan robot lebih presisi dibandingkan dengan sensor depan, akan tetapi dikarenakan penggunaan robot ini pada ruangan labirin kecil yang tidak memiliki jarak yang terlalu jauh, maka error pada sensor masih bisa diterima dan sensor ini layak digunakan untuk digunakan pada robot berkaki.

C. Pengujian dan Analisa Servo.

Komunikasi serial dilakukan antara Arduino Mega dengan servo AX-12 atau AX-18. Pada dasarnya di dalam servo AX-12 dan servo AX-18 sudah memiliki kontroler yang berada di dalamnya. Sehingga kita cukup membuat perintah yang dikirimkan dari Arduino Mega ke dalam servo AX-12 dan AX-18. Pengujian dilakukan apakah servo dapat bergerak sesuai dengan program yang diberikan. Program menggerakkan servo dan hasil pergerakannya dapat dilihat pada tabel 3.

TABEL 5.
PENGUJIAN GERAK SERVO

Perintah	Pergerakan Servo
walk(0,0,0.0)	Stop
walk(10,0,0.0)	Maju
walk(-10,0,0.0)	Mundur
walk(0,10,0.0)	Geser Kiri
walk(0,-10,0.0)	Geser Kanan
walk(0,0,0.1)	Putar Kiri
walk(0,0,-0.1)	Putar Kanan

Pada tabel 3 berisi perintah untuk menggerakkan servo-servo pada kaki robot. Fungsi *walk* tersebut memiliki 3 parameter yaitu x,y,dan z. Parameter x merupakan gerakan *linier* maju, parameter y merupakan gerakan *linier* samping, sedangkan parameter z merupakan gerakan *angular* untuk robot berputar. Nilai yang ada dalam fungsi tersebut dalam satuan cm/s untuk gerak *linier* dan dalam radian/s untuk gerak *angular*.

Pada program putar baik untuk putar kiri maupun putar kanan dengan sudut putar 90 derajat dibutuhkan value 0,31 dalam waktu 2s atau butuh dikirim datanya sebanyak 22 kali. Dengan menggunakan perintah tersebut, dapat dilakukan pemasukan nilai kecepatan pada servo dengan lebih fleksibel. Perintah ini sudah diolah dengan *library* yang tersedia sehingga penumpukan *buffer* dapat diatasi.

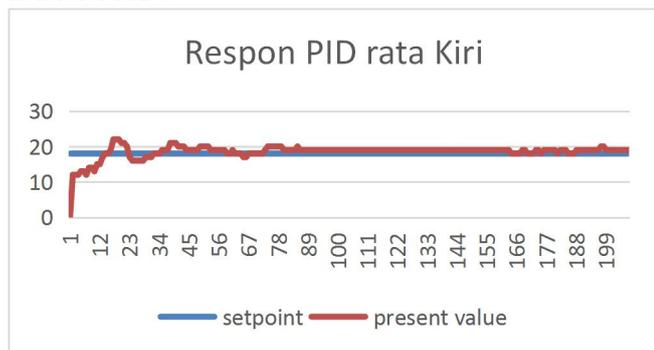


D. Pengujian dan Analisa Kontrol PID.

Pengujian ini dilakukan untuk mengetahui apakah kontrol PID dapat bekerja dengan baik atau tidak. Pengujian ini dilakukan dengan cara memberikan nilai Kp, Ki dan Kd pada sebuah kontrol dengan metode *trial error*. Respon dari sistem ini adalah kecepatan nilai PV untuk mencapai SP dimana nilai PV merupakan nilai dari pembacaan sensor Lidar LDS-01 ke dinding secara langsung, dan nilai SP merupakan nilai jarak antara sensor dengan dinding yang telah di-*setting*. Variabel mode tersebut digunakan sebagai parameter robot akan bergerak menggunakan mode *wall following* kanan atau *wall following* kiri.

E. Pengujian kontrol PID Rata Kiri

Pengujian robot berjalan dengan menggunakan metode *wall following* rata kiri dengan menentukan arah tujuannya dengan nilai Konstanta PID yang telah di-*setting* agar dapat mempertahankan pada nilai *setpoint*. Respon ini didapat saat robot berjalan lurus. Berikut adalah grafik dari kontroler PID :



Gambar 20. Tampilan Pengujian Respon Sinyal Kontroler PID Rata Kiri

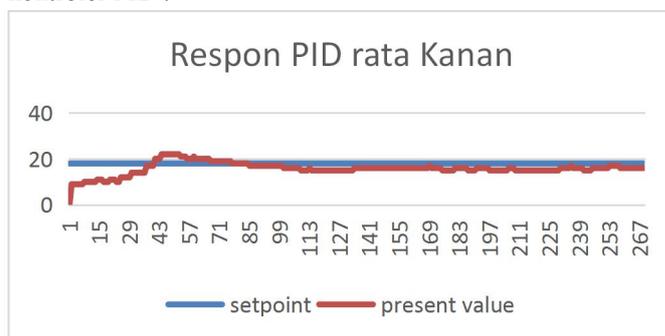
Pada gambar 20 dapat dilihat dan dianalisa bahwa:

1. *Risetime* (T_r) = merupakan lama waktu yang diperlukan respon menuju *setpoint* mulai dari $t = 0$ sampai dengan respon menyentuh sumbu *setpoint* yang pertama nilai $T_r = 170$ ms.
2. *Peaktime* (T_p) = waktu yang diperlukan respon menuju titik puncak pertama dari *overshot*. Pada gambar waktu yang diperlukan adalah 200 ms.
3. Respon signal yang didapat terlihat bahwa saat awal-awal robot masih terjadi osilasi tetapi semakin waktu robot akan mencapai menuju titik *setpoint*.

Dari hasil *trial error* untuk mendapatkan nilai Kp, Ki, dan Kd didapatkan nilai $K_p = 0,3$, $K_i = 0,2$ dan $K_d = 4,5$. Nilai yang didapat tersebut merupakan nilai yang paling mendekati ideal saat robot bernavigasi.

F. Pengujian kontrol PID Rata Kanan

Pengujian robot berjalan dengan menggunakan metode *wall following* rata kanan dengan menentukan arah tujuannya dengan nilai Konstanta PID yang telah di-*setting* agar dapat mempertahankan pada nilai *setpoint*. Respon ini didapat saat robot berjalan lurus. Berikut adalah grafik dari kontroler PID :



Gambar 21. Tampilan Pengujian Respon Sinyal Kontroler PID Rata Kanan

Pada gambar 21 dapat dilihat dan dianalisa bahwa:

1. *Risetime* (T_r) = merupakan lama waktu yang diperlukan respon menuju *setpoint* mulai dari $t = 0$ sampai dengan respon menyentuh sumbu *setpoint* yang pertama nilai $T_r = 410$ ms.
2. *Peaktime* (T_p) = waktu yang diperlukan respon menuju titik puncak pertama dari *overshot*. Pada gambar waktu yang diperlukan adalah 510 ms.
3. Respon signal yang didapat terlihat bahwa saat awal-awal robot masih terjadi osilasi tetapi semakin waktu robot akan mencapai menuju titik *setpoint*.

Dari hasil *trial error* untuk mendapatkan nilai Kp, Ki, dan Kd didapatkan nilai $K_p = 0,2$, $K_i = 0,3$ dan $K_d = 7,5$. Nilai yang didapat tersebut merupakan nilai yang paling mendekati ideal saat robot bernavigasi.

V. PENUTUP

A. Kesimpulan

Dari hasil perancangan dan pengujian yang sudah dilakukan, dapat kesimpulan bahwa sistem yang telah dibuat pada robot berkaki dapat bekerja sesuai dengan perancangan. Berikut ini merupakan kesimpulan yang telah didapat :

1. Robot dapat bernavigasi menggunakan rata kanan dan rata kiri dengan benar
2. Robot dapat menghindari apabila diberi gangguan dan mempertahankan *setpoint*
3. Robot dapat bernavigasi menuju ruang yang telah ditentukan dan berhenti saat mendeteksi api
4. Implementasi penggunaan metode PID pada robot berkaki sesuai dengan yang diharapkan. Dari hasil pengujian nilai $K_p = 0,3$, $K_i = 0,2$, dan $K_d = 4,5$.



Konstanta tersebut dihasilkan respon sistem yang sesuai dengan harapan.

B. Saran

Ada beberapa saran yang dapat digunakan untuk pengembangan robot berkaki dengan sensor lidar diantaranya sebagai berikut :

1. Pengidentifikasian ruang dapat dilakukan dengan cara *training* data sensor untuk dimasukkan ke dalam AI agar dapat memprediksi ruangan tanpa harus robot dalam kondisi lurus dengan dinding ruangan dan masuk ke ruangan.
2. Untuk melakukan navigasi dapat menggunakan metode atau kontrol lain seperti kontroler *fuzzy logic*.
3. Agar dapat lebih mudah untuk logika berjalan, *localitation*, *mapping* apabila dapat menyambungkan robot dengan *package* milik *turtlebot.org*.

Ruchanurucks. "ROS based autonomous mobile robot navigation using 2D LiDAR and RGB-D camera." 2019 First international symposium on instrumentation, control, artificial intelligence, and robotics (ICA-SYMP). IEEE, 2019.

- [12] Li, Yong, and Changxing Shi. "Localization and navigation for indoor mobile robot based on ros." 2018 Chinese Automation Congress (CAC). IEEE, 2018.
- [13] Quigley, Morgan, et al. "ROS: an open-source Robot Operating System." ICRA workshop on open source software. Vol. 3. No. 3.2. 2009

REFERENSI

- [1] Cherubini, A. (2013). Lidar-based Teach-and-Repeat of Mobile Robot Trajectories.
- [2] Suharto, S. B. (2009). Kontrol sudut elevasi robot pelontar softsaucer dengan metode PID.
- [3] Suryatini. (2013). Robot Cerdas Pemadam Api Menggunakan PING Ultrasonic Range Finder dan Uvtron Flame Detector Berbasis Mikrokontroler Atmega 128.
- [4] Al Azhar, Gillang, 2017. "Kontrol sudut elevasi robot pelontar softsaucer dengan metode PID". Skripsi Teknik Elektro Politeknik Negeri Malang.
- [5] Firda, Dwi, 2018. Kontrol Kecepatan Motor Pelempar Pada Robot ABU ROBOCON 2018 Dengan Metode PID
- [6] Sun, Dong, et al. "Global stability of a saturated nonlinear PID kontroler for robot manipulators." *Ieee Transactions On Control Systems Technology* 17.4 (2009): 892-899.
- [7] Buku Panduan Kontes Robot Pemadam Api Indonesia (KRPAI) 2019. Direktorat Penelitian dan Pengabdian kepada Masyarakat, Direktorat Jenderal Pendidikan Tinggi, Kementerian Pendidikan Nasional. Jakarta. 2019.
- [8] Cahyono, Handika Dwi, 2015. "Kontrol Lengan Robot Badminton dan Analisis Kinematic". Skripsi Teknik Elektro Politeknik Negeri Malang.
- [9] Cheng, Yi, and Gong Ye Wang. "Mobile robot navigation based on lidar." 2018 Chinese Control And Decision Conference (CCDC). IEEE, 2018
- [10] Eustice, Ryan M., and Ryan W. Wolcott. "Visual localization within LIDAR maps." *U.S. Patent No. 9,989,969*. 5 Jun. 2018.
- [11] Gatesichapakorn, Sukkpranhachai, Jun Takamatsu, and Miti

