

ANALISA ALGORITMA PROPORSIONAL INTEGRAL DERIVATIF DAN ALGORITMA PATH PLANNING PADA ROBOT LINE FOLLOWER

Mokhammad Aripin¹, Ridho Taufiq Subagio², Deny Martha³

Sekolah Tinggi Manajemen Informatika dan Komputer CIC Cirebon

Jl. Kesambi 202, Kota Cirebon, Jawa Barat. Tlp: (0231)220250.

Email: aripinmokhammad@gmail.com¹, ridho.taufiq@gmail.com², deny.martha@cic.ac.id³

Abstrak

Penjadwalan kuliah merupakan suatu kegiatan penting yang dilakukan setiap awal semester. Proses pembuatan jadwal kuliah di STMIK CIC Cirebon masih dilakukan secara manual, sehingga dibutuhkan waktu yang cukup lama untuk menyelesaikannya. Hal ini dikarenakan beberapa kriteria atau syarat tertentu yang membuat penjadwalan kuliah menjadi rumit, diantaranya kesediaan waktu untuk mengajar dan mata kuliah untuk masing-masing dosen berbeda, juga keberadaan kelas-kelas paralel.

Metode penelitian yang digunakan terdiri dari beberapa tahapan penelitian, yaitu pengumpulan data, analisis sistem dan perancangan perangkat lunak yang akan dibuat. Metode pengumpulan data yang digunakan adalah observasi, dan studi pustaka. Metode analisis dan perancangan sistem yang digunakan adalah Object Oriented Analysis (OOA) menggunakan Unified Modeling Language (UML). Aplikasi penjadwalan kuliah akan dibuat dengan bahasa pemrograman php menggunakan Yiiframework dan MySQL sebagai database-nya, serta menerapkan algoritma genetika untuk menyelesaikan permasalahan penjadwalan.

Hasil dari penelitian ini adalah sebuah aplikasi penjadwalan kuliah yang dapat digunakan oleh staff BAAK untuk membuat jadwal kuliah di Program Studi Informatika. Penerapan aplikasi ini dapat memberikan kemudahan bagi staff BAAK, serta dapat menghemat waktu untuk pembuatan jadwal perkuliahan.

Kata Kunci : Algoritma Genetika, Penjadwalan Kuliah, Yiiframework.

Abstract

Currently, the development of robotics technology has been able to improve the quality and quantity of production of various factories. Robotics technology has also been reaching out in terms of entertainment and education for people. One type of robot that is most attractive is the kind of line follower robot. There are two types of line follower robot is a robot line follower without using the programs (analog) and a line follower robot using a program (digital). Line follower robot using an algorithm that takes the program a robot can run automatically and stable movement.

In this research line follower robot using an algorithm proportional, integral, derivative (PID), algorithm path planning and ATmega32 microcontroller as controller component, where this microcontroller control system of the robot in accordance with the desired program and is written therein. Software (software) in line follower robot was created using the C language with IDE (Integrated Development Environment) CodeVision-AVR® made specifically for AVR-based microcontroller.

Results of the analysis algorithm PID line follower robot that can run stable in low speed and normal, but in the high-speed line follower robot can not run stable. Results of the analysis algorithm robot path planning is set by the user by looking at crossings and determine which path to go through the first robot then determines the intersection by giving orders.

Keyword : Line Follower Robot, Algorithm Proportional Integral Derivative (PID), Algorithm Path Planning

1. PENDAHULUAN

Penerapan mikrokontroler pada bidang robotika terus mengalami perkembangan yang sangat pesat. Banyak negara maju (Amerika, Jerman, Inggris, Jepang, Perancis) berlomba-lomba untuk menciptakan robot-robot cerdas dengan keistimewaan-keistimewaan khusus. Termasuk Indonesia sebagai negara berkembang, hal ini terlihat dengan diselenggarakannya ajang Kontes Robot Indonesia (KRI) dan Kontes Robot Cerdas Indonesia

(KRCI) oleh Departemen Pendidikan Nasional (Depdiknas). Pembuatan robot-robot dengan keistimewaan khusus sangat berkaitan erat dengan adanya kebutuhan dalam dunia industri modern yang menuntut adanya suatu instrument dengan kemampuan yang tinggi yang dapat membantu menyelesaikan pekerjaan manusia ataupun untuk menyelesaikan pekerjaan yang tidak mampu diselesaikan oleh manusia.

Salah satu jenis robot dengan kemampuan keistimewaan yang belakangan ini banyak menarik minat para ahli untuk dikembangkan adalah *mobile robot*. Menurut Dwi Hartanto (2005:182) *Mobile robot* merupakan robot yang dapat berpindah dari tempatnya menuju tempat lain. *Mobile robot* menyerupai fungsi makhluk hidup yang dapat berpindah, jenis robot ini biasanya diciptakan untuk berbagai keperluan, seperti: mengangkut barang secara otomatis, melakukan perjalanan atau pemantauan ke tempat-tempat berbahaya, sebagai alat hiburan (*Robotainment*) atau mainan. Kemampuan dari *mobile robot* sangat beragam sesuai dengan tingkat dan jenis keperluan. Salah satunya adalah robot *line follower* (pengikut garis). Yaitu suatu jenis robot beroda yang memiliki sensor untuk mendeteksi suatu garis dengan pola tertentu kemudian bergerak mengikuti garis tersebut. Robot ini biasa digunakan dalam bidang industri besar sebagai pengangkut barang dari suatu tempat ke tempat lain secara terus-menerus. Misalnya pada perusahaan mobil, robot ini digunakan sebagai pengangkut kerangka mobil yang sudah jadi.

Robot bisa mengetahui keadaan sekitarnya melalui sensor yang dimiliki oleh robot itu sendiri. Ada berbagai sensor yang biasa digunakan pada robot, diantaranya sensor api (*flame detector*), sensor *proximity*, sensor *ultrasonic*, *rotary encoder*, kamera, *magnetic compass*, dan lain-lain. Sensor yang sering digunakan pada *mobile robot* adalah sensor *proximity*, yaitu sensor yang dapat mendeteksi ada atau tidak adanya suatu objek. Pada robot *line follower* sensor *proximity* sering digunakan untuk mendeteksi ada atau tidak adanya suatu garis pembimbing gerak robot. Dengan demikian sensor *proximity* difungsikan sebagai sensor garis. Sensor ini dapat dibuat dari pasangan IRED (*Infra Red Emitting Diode*) dan *phototransistor* yang dipasang secara sejajar dan diterima oleh basis *phototransistor* maka *phototransistor* menjadi saturasi (*off*) sehingga tegangan keluaran mendekati 0 volt, yang didefinisikan sebagai logika '0' atau 'low'. Sebaliknya jika tidak terjadi pantulan, artinya pancaran inframerah dari IRED diserap oleh garis, maka *phototransistor* menjadi *cut-off* dimana tegangan keluaran sama dengan Vcc (5 volt). Kondisi ini didefinisikan sebagai logika '1' atau 'high'.

Prototype robot jenis *line follower* bergerak menggunakan aktuator berupa motor DC. Motor DC merupakan suatu motor penggerak yang dikendalikan dengan arus searah. *prototype* robot jenis *line follower*, memiliki dua buah motor DC yang dipasang di sebelah kiri dan kanan robot, kedua motor bergerak secara diferensial. Untuk dapat menggerakkan robot maka masing-masing motor DC harus diatur arah putar dan kecepatannya lewat sebuah rangkaian driver motor, prinsip kerja dari driver motor yaitu melewatkan arus pada motor dan menghentikan arus yang melewati motor serta mengatur arah arusnya.

Dengan digunakannya algoritma proporsional integral derivatif (PID) pada *prototype* robot jenis *line follower* ini yang bertujuan untuk kestabilan dalam pergerakan robot pada saat robot berjalan dan algoritma *path planning* pada *prototype* robot jenis *line follower* ini yang bertujuan untuk dapat melewati lintasan persimpangan sesuai dengan perintah pengguna dan dimulai dari perintah pertama.

Berdasarkan latar belakang tersebut penulis tertarik untuk mengetahui cara kerja algoritma PID dan algoritma *path planning* pada *prototype* robot jenis *line follower* ini. Penulis membuat penulisan ini dengan judul "Analisa Algoritma Proporsional Integral Derivatif dan Algoritma Path Planning pada Robot Line Follower".

1.1. Identifikasi Masalah

Berdasarkan latar belakang masalah tersebut, maka penulis menguraikan permasalahan yang ada, antara lain :

1. Bagaimana cara kerja algoritma PID pada *prototype* robot jenis *line follower* ?
2. Bagaimana cara kerja algoritma *path planning* pada *prototype* robot jenis *line follower*?

1.2. Batasan Masalah

Adapun batasan masalah yang dibahas adalah sebagai berikut :

1. Algoritma yang digunakan untuk mendapatkan hasil pergerakan yang stabil pada *prototype* robot berjenis *line follower* adalah algoritma PID sebagai pengendali pergerakan pada *prototype* robot berjenis *line follower*.
2. Algoritma yang digunakan untuk melewati lintasan persimpangan pada *prototype* robot berjenis *line follower* adalah algoritma *path planning*. Cara kerja algoritma *path planning* yaitu mengetahui lintasan yang akan digunakan, menentukan jalur mana yang akan dipilih, pengguna mengatur perintah, kemudian *prototype* robot berjenis *line follower* dijalankan.

3. Komponen elektronik yang digunakan pada *prototype* robot berjenis *line follower* yaitu motor dc, sensor, mikrokontroller, battery, LCD, button, *driver* motor.
4. Bahasa pemrograman yang digunakan pada *prototype* robot berjenis *line follower* adalah bahasa C.
5. *Prototype* robot berjenis *line follower* ini menggunakan 10 sensor *photodiode* yaitu 8 sensor pada bagian depan berfungsi untuk membaca lintasan tanpa persimpangan, dan 1 sensor pada bagian samping kanan robot dan 1 sensor pada bagian samping kiri robot berfungsi untuk membaca lintasan persimpangan menggunakan komparator.

1.3. Tujuan

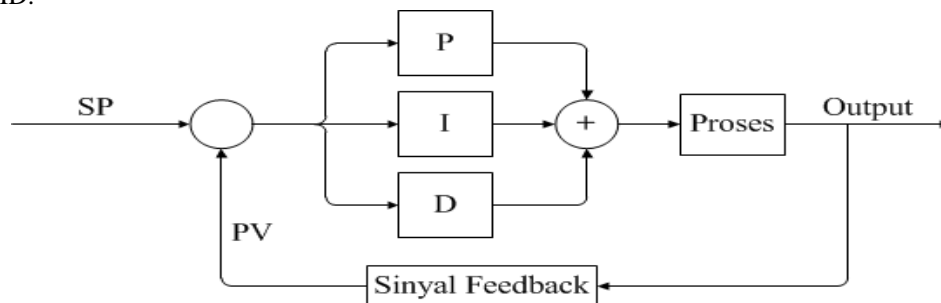
Tujuan dari pembuatan aplikasi ini adalah :

1. Untuk mengetahui cara kerja algoritma PID yang ada pada *prototype* robot jenis *line follower* ini.
2. Untuk mengetahui cara kerja algoritma *path planning* yang ada pada *prototype* robot jenis *line follower* ini.

2. KAJIAN PUSTAKA

2.1. Algoritma Proporsional Integral Derivatif (PID)

Karakteristik kontrol PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter Proporsional (P), Integral (I), dan Derivatif (D). Penyetelan konstanta Kontrol Proporsional (Kp), Kontrol Integral (Ki), Kontrol Derivatif (Kd) akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat diset lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah akan memberikan kontribusi pengaruh pada respon sistem secara keseluruhan. Setiap kekurangan dan kelebihan dari masing-masing Kp, Ki, dan Kd dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler PID.



Gambar 1. Blok Diagram Kontroler PID Loop tertutup (Sumber : Bachri M, 2004)

Dari blok diagram diatas dapat dijelaskan sebagai berikut :

1. SP = *Set Point*, merupakan suatu parameter nilai acuan atau nilai yang kita inginkan.
2. PV = *Present Value*, merupakan nilai bobot pembacaan sensor saat itu atau variabel terukur yang diumpanbalikkan oleh sensor (*sinyal feedback* dari sensor).
3. *Error* = nilai kesalahan, selisih antara variabel terukur atau bobot sensor (PV) dengan nilai acuan (SP). Rumus nilai *error* adalah $error = SP - PV^{[1]}$.

2.2. Algoritma Path Planning

Pergerakan robot yang telah dipertimbangkan biasanya dapat dipisah menjadi dua bagian, yaitu fase penentuan dan fase pelaksanaan. Meskipun pemisahan ini merupakan kunci dari berbagai pendekatan berbasis robot, namun sering dibuat secara explicit dalam Artificial intelligence pada game. Kesederhanaan alami pada lingkungan saat ini adalah salah satu penyebab utama hal tersebut. Sekali direncanakan, pergerakan biasanya diasumsikan untuk mungkin dilaksanakan. (Watkins, 1989)

Path planning merupakan seni untuk menentukan rute mana yang akan diambil, berdasar pada aturan dalam representasi internal pada dataran yang ada saat ini. Path finding adalah suatu pelaksanaan dari rute yang secara teoritis ini, dengan menerjemahkan rencana berdasar representasi internal dalam aturan pada pergerakan dalam lingkungan. Meskipun beberapa aplikasi membutuhkan penggabungan dari dua konsep berikut, perbedaan ini memiliki banyak keunggulan termasuk kesederhanaan, sebuah implementasi modular, dan pengintegrasian yang mudah dari berbagai teknologi.

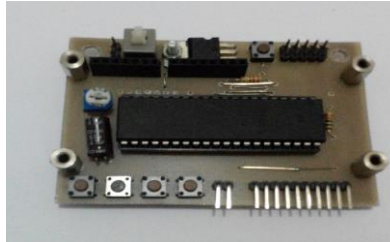
Dan juga, meskipun dua hal ini berbeda dalam konsep, mereka berinteraksi dalam banyak yang terlalu penting untuk dilewatkan. Path planner harus secara alami membiarkan path finder mengetahui path yang sering terpilih. Selanjutnya, harus ada feedback dari path finder kepada path planner sehingga path yang tidak mungkin dicari dapat di catat dan dipertimbangkan^[2].

3. ANALISA SISTEM

3.1. Analisa Perangkat Keras

Analisa perangkat keras yang digunakan pada *prototype* robot jenis *line follower* ini yaitu sistem minimum, sensor photodiode, komparator, motor DC, driver motor, LCD. Berikut adalah penjelasan dari masing-masing perangkat keras.

1) Sistem Minimum

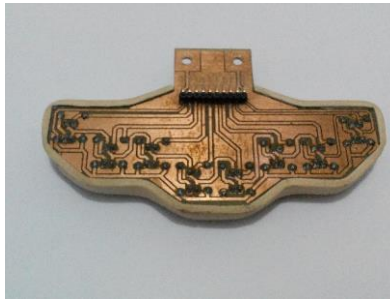


Gambar 2. Sistem Minimum

Keterangan :

Sistem minimum mikrokontroler terdiri dari memori IC ATMEGA32, 4 port yang berfungsi sebagai *sensor*, *driver motor*, *lcd*, dan *komparator*, *reset*, port pendukung untuk memasukkan program kedalam *prototype* robot jenis *line follower*, *power supply*.

2) Sensor Photodiode

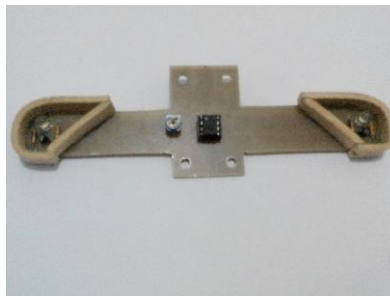


Gambar 3. Sensor Photodiode

Keterangan :

Sensor photodiode yang digunakan adalah sensor dengan 8 bit, yang terdiri dari 8 buah resistor 100 Ω toleransi 5% untuk 8 buah diode, dan 8 buah resistor 330 Ω toleransi 5% untuk 8 buah LED.

3) Komparator

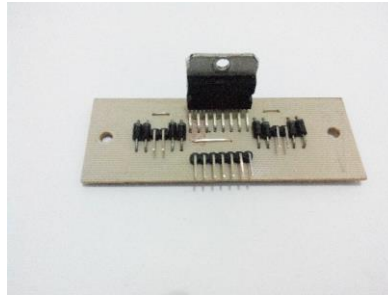


Gambar 4. Komparator

Keterangan :

Komparator terdiri dari IC LM358N dan 2 buah sensor. 1 buah sensor yang berada di bagian tengah samping kiri dan 1 buah sensor yang berada di bagian tengah samping kanan, 2 buah resistor 100 Ω toleransi 5% untuk 2 buah diode, dan 2 buah resistor 330 Ω toleransi 5% untuk 2 buah LED dan trimpot untuk mengatur kepekaan cahaya.

4) Driver Motor



Gambar 5. Driver Motor

Keterangan :

Driver motor terdiri dari IC L298N, 7 pin untuk menghubungkan driver motor ke mikrokontroler, 2 pin untuk menghubungkan motor sebelah kanan, dan 2 pin untuk menghubungkan motor sebelah kiri.

5) Gearbox

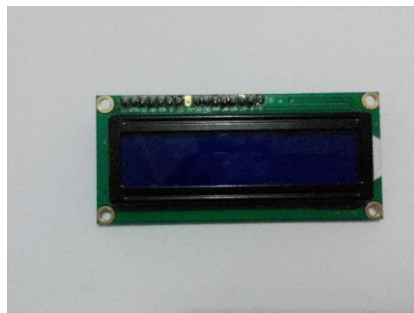


Gambar 6. Gearbox

Keterangan :

Gambar diatas merupakan 1 set gearbox terdiri dari 2 buah roda dan 2 buah motor dc, battery LI-PO dengan tegangan 11,1V (3Cell), akrilik sebagai bahan dasar gearbox.

6) LCD



Gambar 7. LCD

Keterangan :

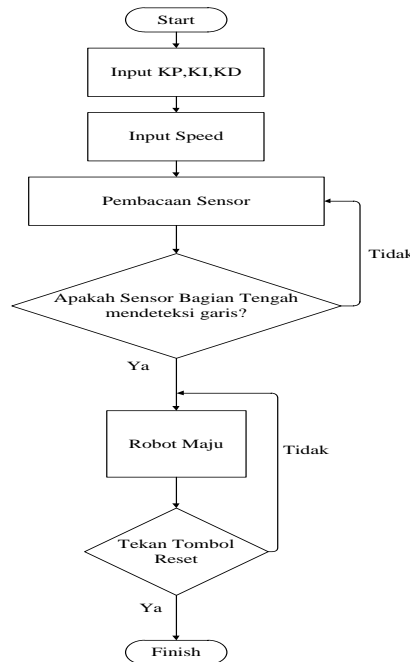
LCD yang digunakan adalah LCD 2x16 berwarna biru. LCD berperan sebagai tampilan agar pengguna dapat mengelola semua menu yang ada pada *prototype* robot jenis *line follower*. Menu diantaranya adalah mengecek sensor, mengatur sensor, mengatur nilai PID, speed, mengatur *path planning*.

3.2. Flowchart PID

Keterangan :

Langkah pertama memberikan nilai Kontrol Proportional (K_p), Kontrol Integral (K_i), Kontrol Derivatif (K_d) pada *prototype* robot *line follower*. Pemberian nilai K_p , K_i , dan K_d berfungsi agar robot dapat berjalan sesuai dengan data yang dibaca pada sensor dan diolah berdasarkan perhitungan K_p , K_i , K_d . Kemudian memberikan nilai *speed* untuk mengatur kecepatan pada robot saat berjalan melintasi perlintasan. Sesudah pemberian nilai, robot melakukan pembacaan sensor. Pembacaan sensor dilakukan sampai robot berada dalam posisi terbaik, dalam penelitian ini posisi terbaik pada robot adalah berada di bagian tengah sensor yang bernilai

1110011 yang artinya robot akan terus berjalan stabil, lurus mengikuti perlintasan. Robot akan berhenti ketika robot di reset atau di matikan.

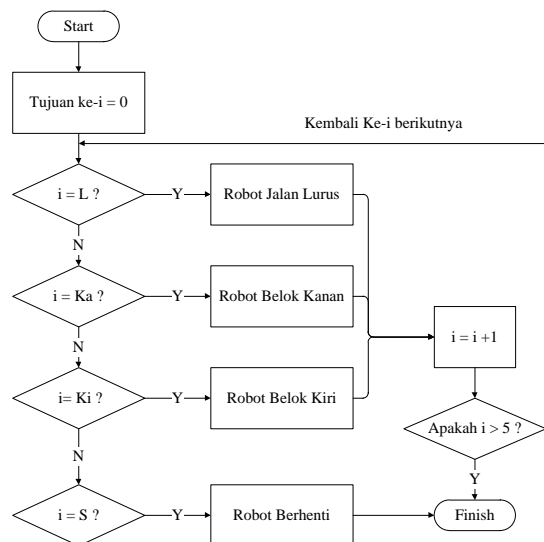


Gambar 8. Flowchart PID

3.3. Flowchart Path Planning

Keterangan :

Flowchart path planning digunakan untuk lintasan persimpangan dimana robot dapat melintasi perlintasan bercabang ketika robot telah disetting menu *path planning* nya. Flowchart diatas menjelaskan bahwa robot mendeteksi tujuan ke-i. apakah i adalah nol jika ya maka robot berjalan lurus jika tidak ke pilihan berikutnya, apakah i adalah 1 jika ya maka robot belok ke kanan jika tidak ke pilihan berikutnya, apakah i adalah 2 jika ya maka robot belok ke kiri jika tidak ke pilihan berikutnya, apakah i adalah 3 jika ya maka robot berhenti. Pada saat robot mendapat nilai 1-3 maka robot akan membaca tujuan berikutnya apabila tujuan berikutnya lebih dari 5 maka robot berhenti dan apabila tujuan kurang dari 5 lalu robot di *setting* berhenti maka robot berhenti, tidak membaca tujuan berikutnya.



Gambar 9. Flowchart Path Planning

4. HASIL ANALISA DAN PEMBAHASAN

4.1 Hasil Analisa Dan Pembahasan

4.1.1. Cara Mengoperasikan Prototype Robot Jenis Line Follower

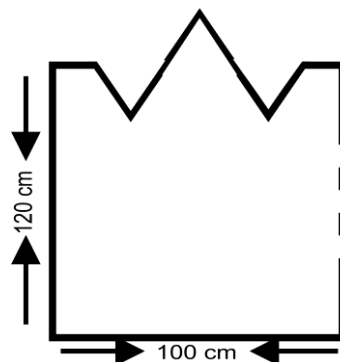
Cara mengoperasikan *prototype* robot *line follower* ini, sebagai berikut :

1. Menghubungkan komponen sistem minimum Atmega32 dengan komponen sensor pada PortA, sensor samping pada Portc.4 dan Portc.5.
2. Menghubungkan komponen sistem minimum Atmega32 dengan motor driver pada Portd.3, Portd.4, Portd.5, Portd.6.
3. Menghubungkan komponen motor driver dengan 2 buah motor DC.
4. Menghubungkan komponen sistem minimum Atmega32 dengan LCD 16x2 ke PortB. Dan tombol switch ke portc.0, portc.1, portc.2, portc.3.
5. Menghubungkan komponen sistem minimum Atmega32 dengan USBAsp.
6. Menghubungkan komponen USBAsp dengan laptop melalui kabel USB.
7. Setelah semuanya terpasang maka CodeVisionAVR dapat dijalankan untuk proses *compile* program.

4.1.2. Lintasan

4.1.1. PID

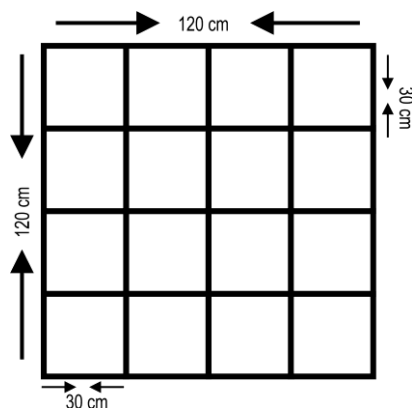
Lintasan *path planning* berbahan dasar flexi dengan latar putih berukuran 120cm x 100cm. lintasan ini digunakan untuk penerapan algoritma *PID*.



Gambar10. Lintasan PID

4.1.2. Path Planning

Lintasan *path planning* berbahan dasar flexi dengan latar putih berukuran 120cm x 120cm dan jarak antara kotak satu ke kotak berikutnya adalah 30cm x 30cm. lintasan ini digunakan untuk penerapan algoritma *path planning*.



Gambar 11. Lintasan Path Planning

4.1.3. Cara Kerja Algoritma PID

Cara kerja algoritma PID pada robot *line follower* ini dilakukan dengan cara *trial and error*. Proses ini dilakukan dengan pemberian nilai secara manual pada proporsional, derivatif, dan integratif pada PID hingga ditemukan hasil sistem yang stabil. Adapun cara yang dilakukan untuk mengatur PID pada robot *line follower* ialah sebagai berikut :

1. Langkah awal gunakan kontrol proporsional (Kp) terlebih dahulu, abaikan kontrol integral (Ki) dan kontrol derivatifnya (Kd) dengan memberikan nilai nol pada integratif dan derivatif.
2. Tambahkan terus nilai Kp maksimum hingga keadaan stabil namun pergerakan robot masih bergelombang.
3. Untuk mengurangi pergerakan yang bergelombang, tambahkan kontrol derivatif (Kd) dengan membagi dua nilai proporsional, amati keadaan sistem robot hingga stabil dan responsif.
4. Jika pergerakan robot telah stabil, pemberian nilai Ki dapat menjadi opsional. Dalam hal ini, penulis tidak menggunakan Ki karena pergerakan robot yang sudah stabil dan pemberian nilai Ki yang tidak tepat dapat membuat pergerakan robot tidak stabil.
5. Nilai kecepatan atau *speed* juga mempengaruhi PID.

Tabel 1. Tabel Hasil Analisa PID

No	Set Point PID				Keterangan		
	Kp	Ki	Kd	Speed	Stabil	Cukup	Kurang
1	0	0	0	100	-	-	√
2	10	0	0	100	-	-	√
3	20	0	0	100	-	-	√
4	30	0	0	100	-	-	√
5	40	0	0	100	-	-	√
6	50	0	0	100	-	√	-
7	50	0	5	100	-	√	-
8	60	0	5	100	-	√	-
9	60	0	10	100	√	-	-
10	60	0	10	115	√	-	-
11	60	0	10	130	-	√	-
12	60	0	20	150	-	-	√

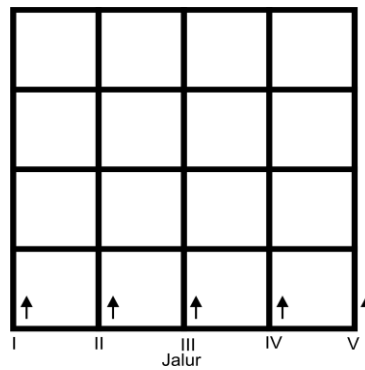
Dari tabel hasil analisa PID dapat disimpulkan bahwa kestabilan *prototype* robot jenis *line follower* ini didapat dengan perhitungan nilai $error = Set\ Point - Present\ Value$. Nilai *error* merupakan selisih antara nilai bobot pembacaan sensor saat itu (*Present Value*) dengan nilai acuan (Set Point). Nilai terbaik untuk kestabilan

pada *prototype* robot jenis *line follower* ini yaitu pada nilai set point K_p adalah 60, K_i adalah 0, K_d adalah 10, dan *speed* antara 100 sampai 120 dengan lintasan pengujian PID yang penulis gunakan. Apabila nilai K_d lebih besar daripada K_p maka pergerakan robot tidak stabil.

4.1.4. Cara Kerja Algoritma Path Planning

Path planning digunakan robot saat melewati lintasan persimpangan berdasarkan perintah pengguna. Apabila ingin menggunakan menu *path planning*, pengguna harus membaca dari arah depan robot bukan dari arah belakang robot.

Ada beberapa model yang dilakukan untuk mendapatkan hasil analisa pada algoritma *path planning*, dari setiap model memiliki perintah, hasil dari perintah, dan hasil pergerakan robot berdasarkan jalur. Perintah yang digunakan adalah kanan (Ka), kiri (Ki), lurus (L), stop (S) dan jalur yang digunakan adalah I, II, III, IV, V. Bagian akhir dari analisa *path planning* adalah mengetahui hasil pergerakan robot berdasarkan masing-masing jalur, apakah sesuai perintah atau tidak.

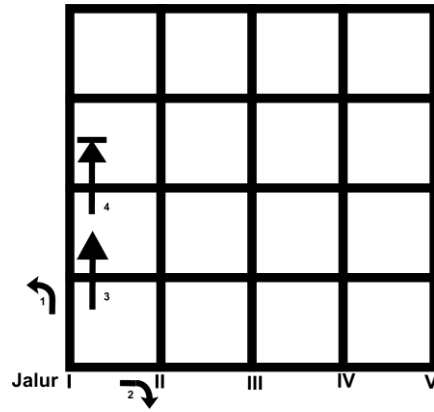


Gambar 12. Jalur Path Planning

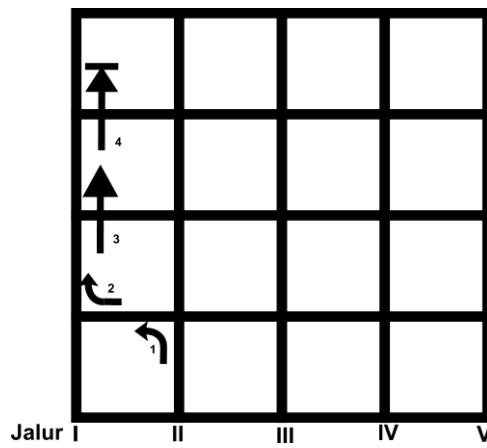
A. Analisa Persimpangan

Tabel 2. Tabel analisa Persimpangan

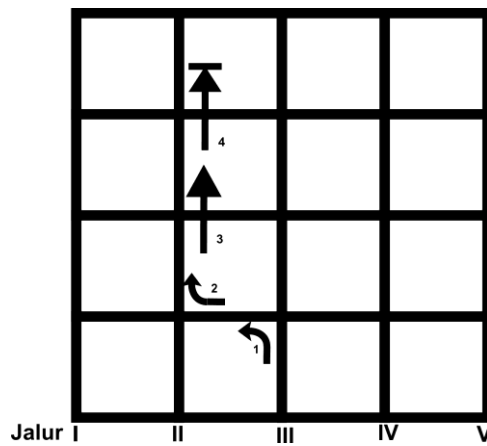
Perintah				Hasil				Pergerakan Robot Pada				
Ka	Ki	L	S	Ka	Ki	L	S	Baris				
								1	2	3	4	5
√	-	-	-	√	-	-	-	√	√	√	√	√
-	√	-	-	-	√	-	-	√	√	√	√	√
-	-	√	-	-	-	√	-	√	√	√	√	√
-	-	-	√	-	-	-	√	√	√	√	√	√
√	-	-	-	-	-	-	-	-	-	-	-	-



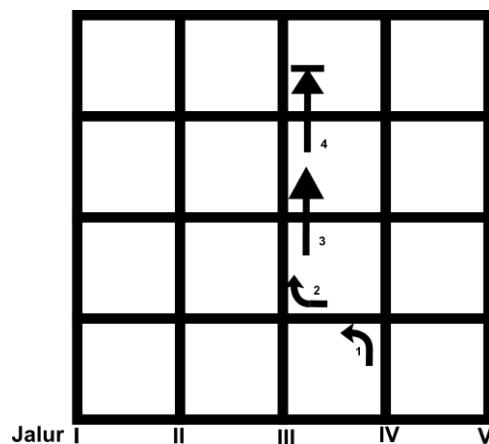
Gambar 13. Pergerakan Robot Pada Jalur I



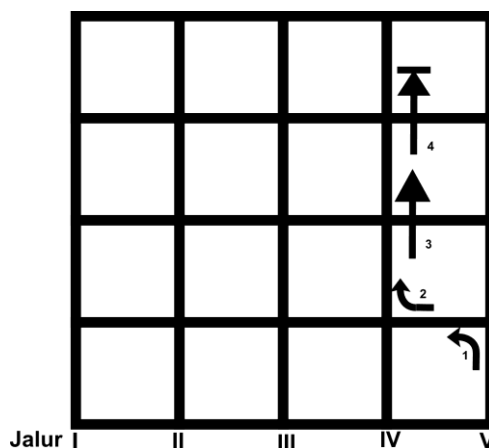
Gambar 14. Pergerakan Robot Pada Jalur II



Gambar 15. Pergerakan Robot Pada Jalur III



Gambar 16. Pergerakan Robot Pada Jalur IV



Gambar 17. Pergerakan Robot Pada Jalur V

Keterangan :

Dari hasil analisa persimpangan, robot memiliki lima perintah yaitu perintah ke satu sampai lima adalah robot belok kanan, kiri, lurus, berhenti, dan kanan. Sehingga hasil pergerakan robot pada jalur ke-I, robot berjalan tidak sesuai dengan perintah yang diharapkan karena pada jalur ke-I untuk perintah ke 2 ketika robot diperintahkan belok kiri robot akan cenderung berjalan lurus karena tidak ada lintasan disebelah kiri. Pada ke-II sampai ke-V robot dapat berjalan sesuai dengan perintah.

5. KESIMPULAN DAN SARAN**5.1. Kesimpulan**

Berdasarkan hasil analisa dan pembahasan pada algoritma proporsional integral derivatif (PID) dan algoritma *path planning* pada robot line follower ini dapat disimpulkan sebagai berikut :

1. Salah satu algoritma untuk mendapatkan kestabilan *prototype* robot jenis *line follower* adalah menggunakan algoritma PID. Nilai set point pada algoritma PID yang terbaik untuk *prototype* robot jenis *line follower* dengan nilai K_p adalah 60, K_i adalah 0, K_d adalah 10, dan speed antara 100 sampai 120.
2. Algoritma *path planning* pada *prototype* robot jenis *line follower* ini digunakan untuk melewati lintasan persimpangan. Proses penggunaannya adalah menentukan dahulu jalur mana yang akan diambil, berdasarkan perintah yang diinginkan kemudian dilaksanakan oleh *prototype* robot jenis *line follower*. *Prototype* robot jenis *line follower* ini dapat berjalan sesuai perintah.
3. Saat ini pergerakan *prototype* robot jenis *line follower* berjalan stabil dalam kecepatan normal sekitar 100 m/s (meter/second) sampai 115 m/s, pada saat kecepatan tinggi yaitu 130 m/s sampai 150 m/s robot tidak dapat berjalan stabil.

5.2. Saran

Dalam penyusunan skripsi teknik informatika ini, penulis menyadari masih terdapat banyak kekurangan. Adapun saran-saran yang dapat penulis berikan adalah :

1. Kode program yang ada pada *prototype* robot jenis *line follower* ini perlu dikembangkan lagi agar *prototype* robot jenis *line follower* dapat berjalan stabil dalam kecepatan tinggi.
2. Algoritma *path planning* pada *prototype* robot jenis *line follower* ini perlu dikembangkan dengan menggunakan algoritma *maze solving*. Cara kerja algoritma *maze solving* yaitu robot membaca lintasan terlebih dahulu sampai mencapai *finish* kemudian menyimpannya kedalam *eprom* dan diproses untuk mencari jalur terdekat, setelah itu robot dijalankan kembali dan robot mencapai finish dengan cepat.

DAFTAR PUSTAKA

- [1]. www.atmel.com/images/doc2503.pdf, diakses 7 juni 2015.
- [2]. <https://fahmizaleeits.wordpress.com/tag/line-follower-berbasis-kendali-pid/>, diakses 7 juni 2015.
- [3]. Soni Ruswanto, Endah Suryawati Ningrum, Irwan Ramli, *Pengaturan Gerak dan Keseimbangan Robot Line Tracer Dua Roda menggunakan PID Controller*, Politeknik Elektronika Negeri Surabaya, 2011.