

## Maze Solver Robot Dengan Kendali PID Berbasis Arduino Uno Dengan Tuning Melalui Smartphone Android

**Arief Rahmadiansyah**

Program Studi S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia  
e-mail : ariefrahmadiansyah@mhs.unesa.ac.id

**M. Syariffudin Zuhri**

Dosen Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia  
e-mail : zuhrie@unesa.ac.id

### Abstrak

*Line Maze Solver* adalah sebuah robot yang menggunakan algoritma *maze solving* untuk menyelesaikan *line maze* yang ada. *Line Maze* merupakan sebuah labirin dengan serangkaian jalur yang terhubung dan berbentuk sebuah garis. Robot ini menggunakan Arduino Uno R3 sebagai mikrokontroler. Mikrokontroler menerima umpan balik dari module sensor TCRT5000 yang berfungsi untuk mendeteksi garis hitam pada lintasan dan juga dalam penelitian ini juga di fungsikan sebagai pendeteksi persimpangan. Robot menggunakan Algoritma tangan kiri sebagai dasar program yang artinya robot ini akan memilih belok kiri di persimpangan dan akan memilih lurus di pertigaan yang hanya memiliki pilhan lurus dan belok kanan . Disetiap Persimpangan robot ini akan merakam setiap jalur yang diambil dan akan memberi kode L(*Left*), R(*Right*), S(*Straight*),B(*Back*). Dan untuk mendapatkan jalur terpendek robot akan menyederhanakan kode dengan menggunakan algoritma *maze solving*, sehingga robot akan jalan langsung menuju finish. Hasil dari penelitian ini adalah robot menggunakan metode *tuning PID trial and error* mendapatkan nilai  $K_p = 10$ ,  $K_i = 30$ ,  $K_d = 45$ . Bila nilai tersebut diubah maka akan berdampak pada sistem follow robot yang mengakibatkan robot salah merekam lintasan. Hasil lain dari penelitian ini robot dapat menemukan jalur yang lebih singkat setelah optimisasi dibandingkan pada awal penelusuran ,hal tersebut menunjukkan bahwa sistem bekerja dengan baik.

**Kata Kunci :** *Line Maze Solver*, PID, TCRT5000, Arduino Uno R3

### Abstract

Line Maze Solver is a robot that uses the maze solving algorithm to complete the line maze. Line Maze is a labyrinth with a bundle of lines that connected and shaped in a line. This robot uses Arduino Uno R3 as a microcontroller. The microcontroller receives feedback from the TCRT5000 sensor module used to detect black lines on the trajectory, and also in this research the functions as a crossing detector. The robot uses the left-hand algorithm as the basis of the program which means this robot will choose to turn left at the intersection and will choose straight at the T-junction which only has a straight choice and turn right. In the maze, this robot will record every path and will give the code L (Left), R (Right), S (Straight), B (Back). To get the shortest path, the robot will simplify the code using the maze solving an algorithm, so that the robot will run directly to the finish. The results of this research the robot is using the trial and error PID tuning method get a value of  $K_p = 10$ ,  $K_i = 30$ ,  $K_d = 45$ . If the value is changed it will have an impact on the following robot system which causes the robot to incorrectly record the track. The other results from this research the robot can find a shorter path after optimization than at the beginning, which means the system works well

**Keywords:** *Line Maze Solver*, PID, TCRT5000, Arduino Uno R3

### PENDAHULUAN

Seiring dengan kemajuan jaman, tidak dapat dipungkiri manusia semakin menginginkan sistem kendali otomatis. Salah satu contoh sistem kendali otomatis adalah robot. Robot adalah sebuah alat

mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu kecerdasan buaatannya.

*Line Follower* merupakan sebuah robot yang memiliki sistem kerja akan berjalan mengikuti suatu garis. Dalam perancangan dan implementasi banyak masalah-masalah yang harus dipecahkan. Masalah-masalah itu adalah sistem penglihatan robot, arsitektur perangkat keras yang meliputi perangkat. Line Follower dapat dikembangkan menjadi suatu robot yang dapat menyelesaikan sebuah *Line Maze* yang disebut *Maze Solver*

Salah satu teknik pengendalian sistem adalah kendali sistem PID (Proporsional Integral Derivative). Sistem kendali ini merupakan kendali yang mempunyai tiga komponen, yaitu Kontrol Proporsional ( $K_p$ ), Kontrol Integral ( $K_i$ ), dan Kontrol Derivative ( $K_d$ ). Tujuan digunakannya sistem kendali pada kerja robot ini yaitu untuk membuat nilai keluaran dan nilai yang diinginkan (referensi) sebaik mungkin, dengan kata lain untuk menghasilkan error sekecil mungkin.

Pada Penelitian kali ini peneliti membahas bagaimana cara merancang robot *maze solver* dengan kontrol PID dan cara kerja dari algoritma *maze solving* agar didapatkan sebuah robot yang dapat memetakan sebuah *line maze* dan meringkas hasil pemetaan agar didapatkan sebuah jalan terpendek menuju *finish*.

## KAJIAN PUSTAKA

### Robot

Menurut Salim, Sandy (2007 : 1) "Robot adalah peralatan elektromekanik atau bio-mekanik, atau gabungan peralatan yang menghasilkan gerakan otonomi maupun berdasarkan gerakan yang diperintahkan. Robot yang menggunakan peralatan komunikasi dimungkinkan untuk dikendalikan oleh manusia, seperti lengan robot yang pengendaliannya dilakukan melalui computer ".Untuk dapat di kategorikan sebagai robot modern sebuah robot memiliki empat karakteristik dasar. Karakteristik dasar tersebut adalah sebagai berikut :

1. Memiliki Sensor. Sensor merupakan suatu peralatan yang berguna untuk mengukur ataupun merasakan sesuatu pada lingkungan di luar robot.
2. Memiliki Sistem Kontrol. Sistem kecerdasan bekerja dengan memproses data masukan berupa keadaan ataupun kejadian yang sedang terjadi dari luar lingkungan..
3. Memiliki Aktuator. Peralatan Mekanik berfungsi untuk membuat sebuah robot dapat melakukan sebuah aksi tertentu dan berinteraksi dengan lingkungannya.
4. Memiliki sumber daya. Sebuah robot memerlukan sumber tenaga untuk menggerakkan komponen-komponen elektrika dan mekanika yang terpasang. Sumber energi pada robot mencakup penyedia tenaga

listrik seperti baterai, dan system pengatur transmisi yang bertugas mengkonversi tenaga listrik sesuai kebutuhan komponen robot tersebut.

Robot dapat dibuat untuk berbagai macam aktifitas, namun sebuah robot harus dibuat dengan tujuan untuk kebaikan manusia

### Maze Solver Robot

Maze Solver robot adalah sebuah robot yang memiliki kemampuan untuk memetakan lintasan yang berupa labirin dan robot ini merupakan sebuah robot beroda yang memiliki lintasan berupa garis atau lintasan dengan dinding. Tujuan dari pembuatan robot ini adalah dapat menemukan jalan menuju finish dengan cepat

### Sistem Kontrol PID

Kontroler PID (*Proporsional Integral plus Derivative*) merupakan sebuah kontroler yang banyak digunakan dalam sistem control di industri. Sinyal keluaran PID didefinisikan.

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

Dengan alih ragam laplace dapat diperoleh fungsi alih kontroler PID sebagai berikut :

$$G_C(s) = K_p + K_D s + \frac{K_i}{s} \quad (2)$$

$$K_p = \frac{\tau_i}{K_x \tau^*} \quad (3)$$

$$K_i = K_p \times \frac{1}{\tau_i} \quad (4)$$

$$K_d = K_p \times \tau_d \quad (5)$$

Keterangan :

$K_p$  : Penguatan Proporsional

$m(t)$  : *Output* kendali

$K_i$  : Penguatan Integral

$e(t)$  : *Error* dalam waktu tertentu

$K_D$  : Penguatan Derivatif

$G_C(s)$  : *Close loop*

$\tau_i$  : Konstanta Waktu Integral

$\tau_d$  : Konstanta Waktu Derivatif

$de(t)$  : Determinan t

$dt$  : waktu

Aspek yang sangat penting dalam desain kontroler PID ialah penentuan parameter kontroler PID supaya sistem *close loop* memenuhi kriteria performansi yang diinginkan. Hal ini disebut juga dengan tuning kontroler Dalam sistem kontrol PID *output* yang diharapkan memenuhi beberapa kriteria, seperti:

- 1). Memiliki *rise time* dan *setting time* yang cepat
- 2). Tidak memiliki *steady state error*; dan
- 3). *Overshoot* sekecil mungkin

### TCRT5000 Module

TCRT5000 adalah komponen elektronika terintegrasi buatan Vishay yang memuat pemancar dan detektor infra merah (*infrared*) dalam satu komponen

terpadu. Konstruksi komponen ini yang kompak diatur sedemikian hingga sumber emisi cahaya infra merah dan komponen sensor / detektonya berada pada arah yang sama, dengan demikian mampu mendeteksi keberadaan objek yang mendekat dengan cara mendeteksi pantulan sinar merah yang terpancarkan dan memantul pada permukaan objek tersebut.

Cahaya infra merah yang diemisikan komponen ini memiliki panjang gelombang 950 nm (nanometer) yang kasat mata. Sifat sinar *infrared* yang tak terlihat ini menguntungkan karena berguna dalam aplikasi sensor tanpa mempengaruhi tampilan sekitarnya, misalnya untuk aplikasi layar sentuh, aplikasi pendeteksi keberadaan objek pada suatu bidang permukaan dsb. Komponen sensor / *detector* adalah sebuah *phototransistor*, kinerja deteksi optimal pada saat objek berada pada jarak 2,5 mm (rentang jarak yang dapat dideteksi antara 0,2 mm hingga 15 mm). *Phototransistor* dilapisi dengan lapisan khusus untuk menahan sinar selain infra-merah untuk meningkatkan akurasi sensor. Rata-rata arus keluaran ( $I_C$ ) adalah 1 mA.



Gambar 1. TCRT5000 Module

(Sumber : <https://www.digikey.co.uk/product-detail/en/vishay-semiconductor-opto-division/TCRT5000/751-1033-5-ND/1681167>)

### Android

Menurut Salbino *android* merupakan sistem operasi berbasis *Linux* yang bersifat *open source* dan dirancang untuk perangkat seluler seperti *smartphone* dan komputer tablet (Salbino 2014). *Android* dikembangkan oleh Android, Inc., dengan dukungan finansial dari *Google* yang kemudian diakuisisi oleh *Google* pada tahun 2005. *Android* resmi di rilis pada tahun 2007, bersamaan dengan dirilisnya *Open Handset Alliance*. Sifat *Android* yang *open source* telah membuat bermuncunya berbagai komunitas pengembang aplikasi untuk menggunakan android sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur baru bagi *Android* pada perangkat yang secara resmi dirilis dengan menggunakan system operasi lainnya.

### Bluetooth HC-05

Bluetooth adalah protokol komunikasi wireless yang bekerja pada frekuensi radio 2.4 GHz untuk pertukaran data pada perangkat bergerak seperti PDA, laptop, HP, dan lain-lain. Salah satu hasil contoh modul Bluetooth yang paling banyak digunakan adalah tipe HC-05. modul Bluetooth HC-05 merupakan salah satu modul Bluetooth yang dapat ditemukan dipasaran dengan harga yang relatif murah. Modul Bluetooth HC-05

terdiri dari 6 pin konektor, yang setiap pin konektor memiliki fungsi yang berbeda - beda. Pada perangkat *Bluetooth* HC-05 menggunakan standarisasi IEEE (*Institute Of Electrical and Electronic*) 802.15.1 yang memiliki karakteristik yaitu:

1. Beroperasi di Frekuensi 2.4 Ghz
2. Jangkauan 0-10 m
3. Kecepatan rata rata 2 MBps



Gambar 2. Bluetooth HC-05

(Sumber: <https://mbed.org/users/edodm85/notebook/HC-05-bluetooth>)

## METODE PENELITIAN

### Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif. Penelitian kuantitatif adalah suatu proses menemukan pengetahuan yang menggunakan data berupa angka sebagai alat menganalisis keterangan mengenai apa yang ingin diketahui. (Kasiram, 2010). Pada penelitian ini menggunakan *Arduino* idea, dan *MIT Inventor* untuk merancang dan mensimulasikan serta membuat program android untuk sistem *tuning* pada android.

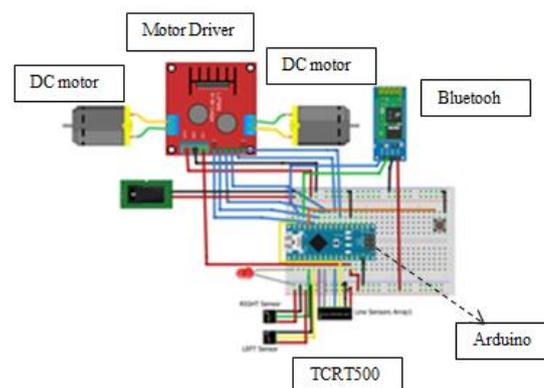
### Tempat dan Waktu Penelitian

Penelitian ini akan dilaksanakan di Laboratorium Sistem Pengaturan dan Kendali Jurusan Teknik Elektro Universitas Negeri Surabaya dan waktu pelaksanaannya dilakukan pada semester genap 2018/2019.

### Rancang Bangun Hardware

Berikut skema rangkaian keseluruhan *hardware* untuk robot Maze Solver berbasis PID

Pada Gambar 5 merupakan gambar yang dibuat

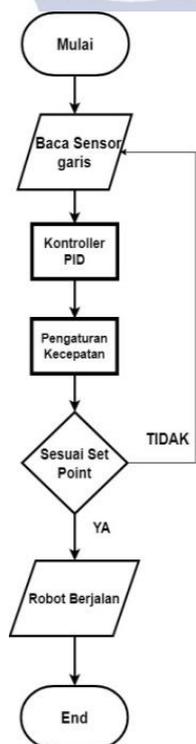


Gambar 3. Desain Hardware  
(Sumber : Data Penelitian, 2018)

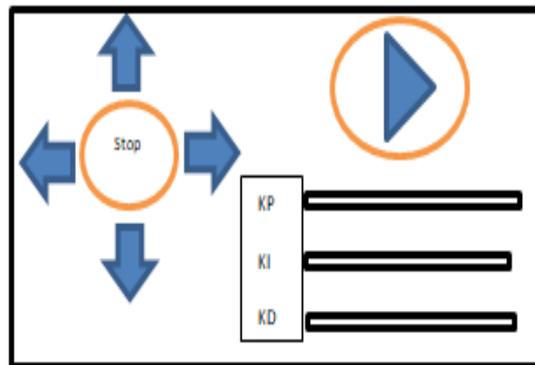
dengan aplikasi fritzing. Pada gambar ini merupakan rangkaian hardware yang akan dibuat terdiri dari Arduino Uno berfungsi sebagai pusat kendali *Maze Solver*, lalu terdapat sensor garis yang berfungsi sebagai pembaca garis warna hitam dan putih dan terdapat dua buah Motor Dc sebagai penggerak dari robot dan satu Motor Driver 1298N sebagai *driver* motor DC. Untuk komunikasi data dengan android digunakan Module Bluetooth HC-05 sebagai penghubung dengan *smartphone android*.

**Rancang Bangun Software**

. Rancang bangun *software* yang dimaksud adalah sebuah program yang dimasukkan ke dalam mikrokontroler Arduino. Pemrograman dibuat menggunakan bahasa pemrogram khusus oleh Arduino pada Arduino IDE (*Integrated Development Environment*) versi 1.6.12. Arduino merupakan bahasa turunan dari C++ sehingga fungsi-fungsi C++ serta C dapat berjalan di Arduino. Pemilihan bahasa Arduino ini dipilih karena untuk memadai perhitungan matematis dalam sistem pengontrolan motor servo dan bersifat *open source*. Karena sifatnya yang *open source*, maka banyak *library* yang dikembangkan oleh personal atau komunitas di luar *developer* Arduino itu sendiri.



Gambar 4. Flowchart Sistem  
(Sumber : Data Penelitian, 2018)



Gambar 5. Desain Aplikasi  
(Sumber : Data Penelitian, 2018)

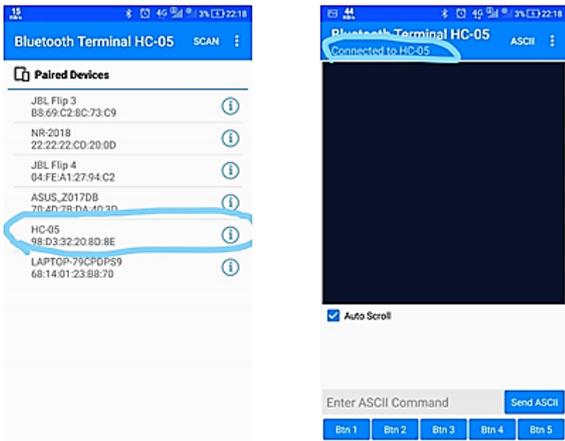
Gambaran umum jalannya program adalah membaca nilai kecepatan. Dan membandingkannya dengan nilai referensi (*setpoint*), kemudian Pada flowchart tersebut dapat diketahui bahwa sistem pengaturan kecepatan motor ini memiliki kemampuan untuk terus menyesuaikan kondisi sistem sampai akhirnya tercapai kondisi tunak atau *Steady state*. Sebagai masukan sistem, setpoint akan diberikan pada pengendali PID. Setpoint dalam hal ini merupakan kecepatan yang akan disesuaikan sehingga saat kecepatannya sudah mencapai posisi tersebut, dapat dikatakan bahwa system telah mencapai keadaan tunak. Pada saat kecepatan belum mencapai keadaan tunak, sensor akan membaca kecenderungan beban yang diemban.

**HASIL DAN PEMBAHASAN**

Hasil dari skripsi ini adalah menghasilkan sebuah robot *maze solver* dengan kontrol PID . Pada bagian , pengujian *bluetooth*, pengujian sensor, control PID, sistem *maze solver*, dan aplikasi robot di android.

**Pengujian Konektivitas Bluetooth**

Pada robot *maze solver* ini digunakan *bluetooth* HC-05 sebagai penghubung ke aplikasi *android*. Pada pengujian kali ini saya menggunakan aplikasi android HC-05 terminal sebagai pembuktian bahwa *bluetooth* telah terhubung dengan perangkat android dan juga dengan aplikasi ini saya menggunkannya untuk mengukur jarak konektivitas *bluetooth*.



Gambar 6 Pengujian konektivitas antara robot dengan perangkat android menggunakan aplikasi HC-05 terminal

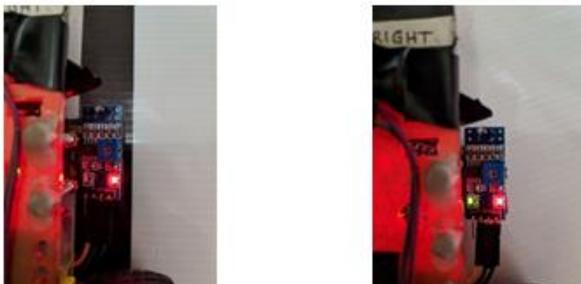
Tabel 1. Hasil pengujian jarak jangkauan alat

NO	Jarak	Status
1	1 m	Terhubung
2	2 m	Terhubung
3	3 m	Terhubung
4	4 m	Terhubung
5	5 m	Terhubung
6	6 m	Terhubung
7	7 m	Terhubung
8	8 m	Terhubung
9	9 m	Terhubung
10	10 m	Terhubung
11	11 m	Tidak Terhubung

(Sumber : Data Penelitian, 2019)

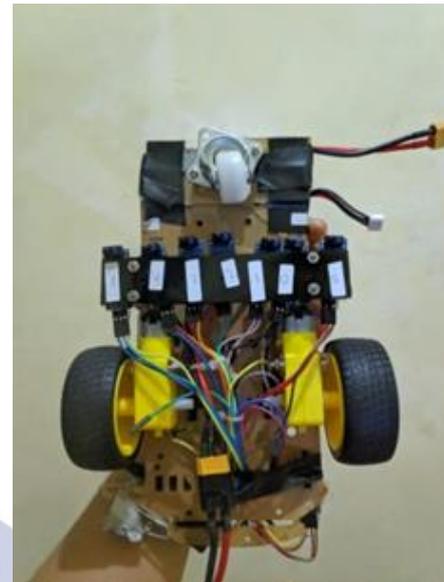
### Pengujian Sensor TCRT5000

Pada bagian ini akan menguji sensor TCRT5000. Output yang digunakan sensor adalah output digital jadi jika sensor mendeteksi garis putih akan bernilai 0 dan lampu led kuning dan merah menyala, sementara jika mendeteksi hitam akan bernilai 1 dan lampu led merah akan menyala dan kuning mati.



Hitam deteksi 1 Putih deteksi 0

Gambar 7 Pendeteksian sensor (Sumber : Data Penelitian, 2019)



Gambar 8 setup peletakan sensor sensor 6 (sensor kanan), sensor 5(sensor kiri), dan sensor 4, 3, 2, 1, 0 untuk mendeteksi garis lurus

Tabel 2. Fungsi Sensor

No	Sensor	Fungsi
1	0	Sebagai penjejak garis lurus dan penambah nilai probabilitas di persimpangan
2	1	Sebagai penjejak garis lurus dan penambah nilai probabilitas di persimpangan
3	2	Sebagai penjejak garis lurus dan penambah nilai probabilitas di persimpangan
4	3	Sebagai penjejak garis lurus dan penambah nilai probabilitas di persimpangan
5	4	Sebagai penjejak garis lurus dan penambah nilai probabilitas di persimpangan
6	5	Pendeteksi belok kiri
7	6	Pendeteksi belok kanan

(Sumber : Data Penelitian, 2019)

Tabel 2 menunjukkan fungsi-fungsi sensor didalam tabel tersebut dikatakan bahwa sensor sebagai penambah probabilitas di persimpangan. Hal tersebut dikarenakan robot maze ini menggunakan fungsi probabilitas di program Arduino untuk membedakan setiap persimpangan. Fungsi tersebut dinyatakan sebagai berikut ;

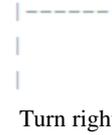
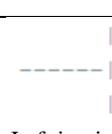
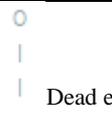
$$\text{probability} = \text{prob}[0] + \text{prob}[1] + \text{prob}[2] + \text{prob}[3] + \text{prob}[4]; \quad (6)$$

Keterangan :

- prob[0] : Nilai Probabilitas sensor 0
- prob[1] : Nilai Probabilitas sensor 1
- prob[2] : Nilai Probabilitas sensor 2
- prob[3] : Nilai Probabilitas sensor 3
- prob[4] : Nilai Probabilitas sensor 4

Jika sensor membaca 1 maka nilai probabilitasnya 0.2 dan jika sensor membaca 0 maka nilai probabilitasnya 0.

Tabel 3. Percobaan gerakan dan syarat pengambilan keputusan

NO	Persimpangan	Syarat	Keputusan	Percobaan
1	 Turn left	5 = 1, 6=0, dan prob: >=0.4	Belok kiri	Berhasil
2	 Turn right	5 = 0, 6=1, dan prob: >=0.4	Belok kanan	Berhasil
3	 T-junction	5 = 1, 6=1, dan prob: >=0.4	Belok Kiri	Berhasil
4	 Left junction	Prob > 0, dan prob < 1	Belok Kiri	Berhasil
5	 Right junction	Prob > 0, dan prob < 1	Lurus	Berhasil
6	 Crossroad	Prob > 0, dan prob < 0.6	Belok Kiri	Berhasil
7	 Dead end	0 = 0, 1=0, 2=0, 3=0, 4=0	Putar balik	Berhasil
8	 Finish	Prob > 0.6, dan prob <= 1	Berhenti	Berhasil

(Sumber : Data Penelitian, 2019)

**Pengujian PID**

Perancangan kontroler PID dilakukan untuk mendapatkan parameter KP, KI, KD. Metode yang digunakan adalah metode *trial and error* dengan cara memasukan nilai secara acak, dimulai dengan memasukan nilai KP lalu KD dan KI.

Tabel 4. Tuning PID

NO	KP	KD	KI	Status
1	1	0	0	Gagal
2	1	1	0	Gagal
3	1	1	1	Gagal
4	5	3	4	Gagal
5	9	20	15	Gagal
6	10	45	30	Berhasil

(Sumber : Data Penelitian, 2019)

Data pada Tabel diatas hanya sebagai sampel yang mewakili dari hasil percobaan secara keseluruhan, setiap percobaan dipastikan menggunakan perlakuan yang sama untuk mendapatkan data yang akurat, seperti pengecekan daya dan kondisi robot secara berkala ketika pengujian serta posisi start dan finish yang sama pada setiap pengujian.

**Pengujian Maze Solver**

Setelah robot dapat membedakan keputusan yang diambil maka tahap selanjutnya adalah robot menyimpan *path* yang telah diambil, lalu menyederhanakan map agar ketika robot dijalankan dapat langsung menuju garis finish.

Untuk dapat menyederhanakan *path* maka dibuat aturan sebagai berikut :

Tabel 5 Penyederhanaan Path

No	Path	Penyederhanaan
1	LBR	B
2	LBS	R
3	RBL	B
4	SBL	R
5	SBS	B
6	LBL	S

(Sumber : Data Penelitian, 2019)

Robot berada di posisi *start 1* maka *path mapping* akan menjadi LLRBLLRLL. Maka jika di *optimize* menggunakan aturan diatas akan menjadi sebagai berikut :

Path = LLRBLLRLL

Penyelesaian = LL(RBL=B)LRLL

LLBLRLL

L(LBL=S)RLL

LSRLL

Maka optimisasi dari *path* LLRBLLRLL menjadi LSRLL

Robot berada di posisi *start 2* maka *path mapping* akan menjadi SLLSRBLLRBLRRL . Maka jika di *optimize* menggunakan aturan diatas akan menjadi sebagai berikut :

Path = SLLSRBLLRBLRRL

Penyelesaian = SLLS(RBL=B)L(RBL=B)LRRL

SLLSBLRRL  
SLL(SBL=R)BLRRL  
SLLRBLRRL  
SLL(RBL=B)RRL  
SLLBRRL  
SL(LBR=B)RLL  
SLBRLL  
S(LBR=B)LL  
SBLL  
(SBL=R)L  
RL

Maka optimisasi dari *path* SLRSLBLLRBLRRL menjadi RL.

Robot berada di posisi *start 3* maka *path mapping* akan menjadi LLBSLLSRBLLRBLRRL. Maka jika di *optimize* menggunakan aturan diatas akan menjadi sebagai berikut :

Path = LLBSLLSRBLLRBLRRL

Penyelesaian=

L(LBS=R)LLS(RBL=B)L(RBL=B)LRRL  
LRLRBLRRL  
LRL(SBL=R)BLRRL  
LRLRBLRRL  
LRL(RBL=B)RRL  
LRLBRRL  
LRL(LBR=B)RLL  
LRLBRLL  
LR(LBR=B)LL  
LRBLL  
L(RBL=B)L  
(LBL=S)  
S

Maka optimisasi dari *path* LLBSLLSRBLLRBLRRL menjadi S

. Robot berada di posisi *start 4* maka *path mapping* akan menjadi LBLBLLSRBLLRBLRRL . Maka jika di *optimize* menggunakan aturan diatas akan menjadi sebagai berikut :

Path = LBLBLLSRBLLRBLRRL

Penyelesaian=

(LBL=S)(LBS=R)LS(RBL=B)L(RBL=B)LRRL  
SRLSBLRRL

SRL(SBL=R)BLRRL

SRLRBLRRL

SRL(RBL=B)RRL

SRLBRRL

SRL(LBR=B)RLL

SRLBRLL

SR(LBR=B)LL

S(RBL=B)L

SBL

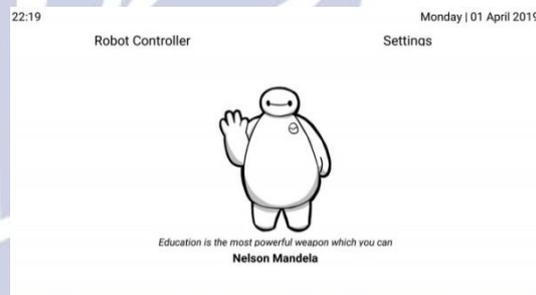
(SBL=R)

R

Maka optimisasi dari *path* LBLBLLSRBLLRBLRRL menjadi R

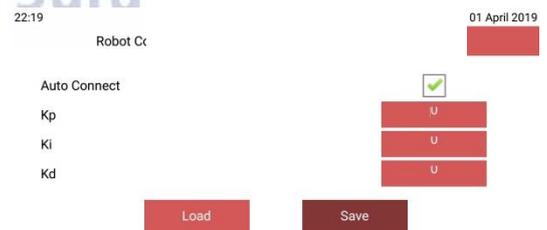
### Aplikasi Android

Aplikasi ini dibuat dengan MIT Invenor APP yang merupakan sebuah aplikasi untuk membuat sebuah aplikasi android dengan sistem *ladder*. Untuk penggunaan aplikasi tersebut pastikan bahwa anda telah mengaktifkan bluetooth pada perangkat android .



Gambar 9 Tampilan Awal (Sumber : Data Penelitian, 2019)

Saat pertama membuka aplikasi tampilan awal pada aplikasi kontrol robot seperti pada gambar 13. Di tampilan awal terdapat dua buah *tab* pilihan yaitu settings dan robot controller.



Gambar 10 Tampilan *tab settings* setelah berhasil pair dengan robot (Sumber : Dokumen pribadi, 2019)

Gambar 12 adalah tampilan ketika kita berhasil melakukan *pair* dengan robot. Pada tab ini terdapat kolom untuk mengisi nilai KP, KI, KD dan terdapat dua buah tab *load* dan *save*. Fungsi pada kedua tab ini adalah tombol *save* berfungsi untuk menyimpan angka dan mengirimkan nilai kp, ki, dan kd ke robot, sedangkan tombol *load* berfungsi ketika kita ingin menggunakan nilai kp, ki, dan kd yang kita gunakan terakhir kali, kita hanya perlu menekan tombol *load* dan nilai kp, kd, dan kd akan terisi secara otomatis.



Gambar 11 Tampilan tab robot controller  
(Sumber : Dokumen pribadi, 2019)

Pada gambar diatas merupakan tampilan awal tab robot controller, tab ini akan aktif bila kita sudah mengisi tab settings. Pada tab ini berisi perintah *forward*, *left*, *stop*, *right*, *backward*, *mapping*, dan *optimize*. Perintah yang aktif hanya perintah yang berwarna merah terang, sedangkan perintah *optimize* belum aktif, perintah ini akan aktif bila robot sudah mulai melakukan.

## PENUTUP

### Simpulan

Berdasarkan penelitian dan pengujian yang dilakukan, maka dapat disimpulkan perancangan robot ini menggunakan Arduino Uno R3 sebagai mikrokontroler dan menggunakan 7 buah sensor TCRT5000 sebagai pendeteksi garis dan pendeteksi persimpangan. Menggunakan 2 buah motor DC sebagai sistem penggerak dan Bluetooth HC-05 sebagai penghubung ke *smartphone* yang digunakan sebagai aplikasi tuning robot.

Kontroler PID yang dirancang pada robot menggunakan metode *trial and error* berdasarkan tabel 4 mendapatkan nilai  $kp = 10$ ,  $ki = 30$ , dan  $kd = 45$ . Nilai tersebut di dapat setelah melakukan percobaan dengan melihat gerak robot dengan memasukkan nilai kp, lalu di ikuti dengan kd dan ki

Algoritma pada maze solver yang diaplikasikan pada robot ini adalah menghilangkan path B yang telah dipelajari oleh robot, path B terus dihilangkan hingga menghasilkan suatu path tanpa path B didalamnya.

Contoh jika robot telah berhasil melakukan *mapping* dan didapat map LLRBLLRLL maka optimisasi dari map tersebut adalah LSRL

### Saran

Berdasarkan hasil penelitian yang telah dilakukan terdapat beberapa aspek yang dapat diperbaiki dan dikembangkan agar memberikan hasil yang lebih maksimal agar menggunakan *motor continuous* servo agar ketika bertemu persimpangan, tikungan dan jalan buntu dapat berputar sesuai derajat.

Menggunakan ban yang memiliki kualitas bagus agar tidak mudah slip saat robot berjalan, karena selama percobaan peneliti memiliki masalah ban slip saat robot berhenti ketika di persimpangan. Dan juga penelitian ini dapat dikembangkan juga dengan menggunakan metode kontroler yang lain seperti *PID-Fuzzy*, *Model Predictive Control*, dan yang lainnya.

## DAFTAR PUSTAKA

- Helie, Robert, 2005. *IEEE Standard for Information Telecommunication and Information Exchange Between System Local and Metropolitan area Network Specific Requirements*. New York : IEEE Computer Society.
- Jatmiko, Wisnu dan dkk. 2012. *Robotika : Teori dan Aplikasi*. Depok. Universitas Indonesia
- Kasiram, Mohammad. 2010. *Metodelogi Penelitian : Kualitatif-Kuantitatif*. Malang : Uin-Maliki Press
- Mishra, Swati. 2008. "Maze Solving Algorithm for Micro Mouse". Makalah disajikan pada IEEE International Conference on Signal Image Technology and Internet Based Systems, Bali, 30 Nopember- 3 Desember
- Ogata, Katsuhiko. 1996. *Teknik Kontrol Automatik*. Jilid I edisi kedua, Terjemahan Ir. Edi Laksono. Jakarta: Erlangga.
- Salbino, Sherief. 2014. *Buku Pintar Gadget Android untuk Pemula*. Jakarta : Kunci Komunikasi.
- Sandy, Salim. 2007. *Merancang Mobile Robot Pembawa Object Menggunakan OOPic-R*. Jakarta : PT. Flex
- Yanto, Febi dan Irma Welly. 2015 "Analisa dan Perbaikan Algoritma *Line Maze Solving* Untuk Jalur *Loop*, Lancip, dan Lengkung Pada Robot *Line Follower* (LFR)". *Jurnal CoreIT* Vol 1(2) : hal 57-62