

# IMPLEMENTASI ALGORITMA *PLEDGE* PADA ROBOT LAPANGAN DALAM PENCARIAN JALAN KELUAR LABIRIN

Yasser Farid S<sup>[1]</sup>, Sarwoko, Ir., M.Sc.<sup>[2]</sup>, Junartha Halomoan, ST., MT<sup>[3]</sup>

Jurusan Teknik Elektro, Institut Teknologi Telkom

## Abstrak

Algoritma *Wall Following* adalah algoritma yang paling sederhana untuk menyelesaikan sebuah labirin. Algoritma *wall following* ini memiliki kelemahan yaitu tidak dapat menemukan solusi sebuah labirin yang memiliki dinding terpisah atau biasa disebut *loop*.

Agar dapat mengatasi masalah di atas, akan digunakan algoritma *Pledge* yang akan diterapkan pada sebuah robot agar dapat memberikan solusi pada labirin yang terdapat *loop* didalamnya. Agar dapat bergerak dengan baik didalam labirin, robot akan diberikan pengontrol dengan metode PID. Kontrol ini akan mengendalikan kecepatan motor DC robot dengan mengkalkulasi nilai *error* berdasarkan *input* dari 3 ultrasonik.

Keluaran sistem yang dicapai adalah robot dapat menemukan solusi jalan keluar sebuah labirin yang terdapat *loop* didalamnya dengan posisi start awal yang berbeda-beda dengan tingkat keberhasilan 100%. Performansi terbaik sistem terdapat pada saat nilai konstanta PID sebesar  $P=70$ ,  $I=0.05$  dan  $D=30$  ditunjukkan dengan nilai performansi sistem  $M_p=7\%$ ,  $t_r=4100\text{ms}$ ,  $t_d=2800\text{ms}$ ,  $t_p=4500\text{ms}$  dan  $t_s=7600\text{ms}$ .

**Kata kunci:** Algoritma *Pledge*, Robot lapangan, Labirin, kontroler PID

## 1. PENDAHULUAN

Perkembangan ilmu pengetahuan memacu perkembangan teknologi yang dapat membantu aktivitas manusia. Pada umumnya manusia tertarik dengan sesuatu yang dapat meringankan beban dari pekerjaannya. Salah satu caranya dengan adanya pemakaian robot.

Pencarian solusi jalan keluar sebuah labirin merupakan salah satu aplikasi robot yang sering digunakan. Dengan menggunakan algoritma yang paling sederhana yaitu *wall follower*, sebuah robot dapat menemukan jalan keluar dari sebuah labirin yang sederhana. Algoritma *wall follower* memiliki keterbatasan dalam mencari jalan keluar dari sebuah labirin jika terdapat *loop* didalamnya. Dengan demikian digunakanlah suatu algoritma sehingga dapat mengatasi kekurangan *wall follower* yaitu Algoritma *Pledge*.

Pada sistem kerja robot juga dibutuhkan suatu input berupa sensor yang berguna sebagai media penginderaan pada robot. Sensor ini juga memungkinkan robot dapat mengetahui posisi robot tersebut terhadap suatu lintasan (apakah robot dalam posisi yang tepat). Posisi robot yang miring dapat disempurnakan dengan dilakukannya kontrol kecepatan motor terhadap perbandingan pembacaan ultrasonik dengan menggunakan kontrol PID.

Adanya error pada posisi robot yang didapat dari sensor ultrasonik akan diperbaiki oleh kontroler PID. Metode ini diharapkan mampu mendapatkan hasil yang terbaik dalam memposisikan robot dalam lintasannya secara tepat dikarenakan banyak kemungkinan yang terjadi pada posisi robot saat melalui lintasan.

## 2. LANDASAN TEORI

### 2.1 Algoritma *Pledge*

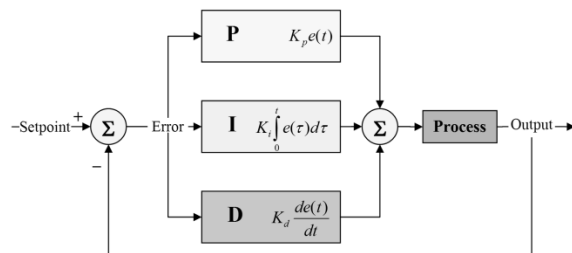
Algoritma yang ditanamkan pada robot ini merupakan salah satu dari *maze solving algorithm* yaitu algoritma *Pledge*. Pada algoritma *Pledge* ini, robot akan terus berjalan lurus pada arah awal yang telah ditentukan jika tidak menemui hambatan apapun. Ketika menemui hambatan robot akan berbelok dan melakukan *wall following* yang akan menghitung jumlah belokan yang ditempuh. Robot akan berjalan lurus tanpa melakukan *wall following* saat kembali kearah awal dengan total belokan adalah 0.

Berikut adalah pseudocode Algoritma *Pledge*.

```
Total_belokan = 0
While (robot menyala)
{
    Berjalan ke arah utama
    If (bertemu halangan)
    {
        Repeat
            Berjalan dengan metode wall
            follower Menghitung total
            belokan yang di tempuh (belok
            kanan Total belokan + 1, belok
            kiri Total belokan - 1)
        Until
            Total_belokan = 0
    }
}
```

## 2.2 Kontrol PID

Di dalam suatu sistem kontrol kita mengenal adanya beberapa macam aksi kontrol, diantaranya yaitu aksi kontrol *proporsional*, aksi kontrol *integral* dan aksi kontrol *derivative*. Masing-masing aksi kontrol ini mempunyai keunggulan tertentu, dimana aksi kontrol *proporsional* mempunyai keunggulan *rise time* yang cepat, aksi kontrol integral mempunyai keunggulan untuk memperkecil *error* dan aksi kontrol *derivative* mempunyai keunggulan untuk memperkecil *error* atau meredam *overshot/undershot*. Untuk itu agar kita dapat menghasilkan output dengan *risetime* yang cepat dan *error* yang kecil kita dapat menggabungkan ketiga aksi kontrol ini menjadi aksi kontrol PID.

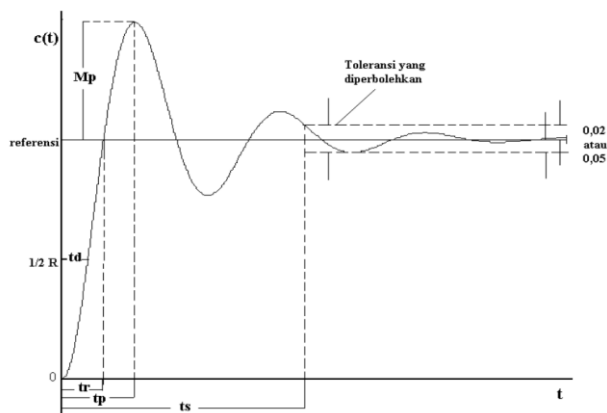


Gambar 2.1 Blok Diagram Kontroler PID

## 2.3 Spesifikasi Performansi Sistem

Tanggapan transient merupakan suatu hal yang penting dalam berbicara masalah sistem. Tanggapan transient suatu sistem control secara praktek selalu menunjukkan osilasi teredam sebelum mencapai keadaan tunaknya. Dalam menggolongkan karakteristik tanggapan transien suatu sistem Kontrol, umumnya dapat dikelompokkan sebagai berikut :

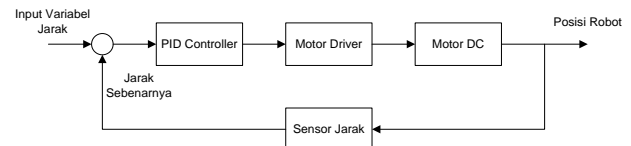
1. Waktu tunda ( $t_d$ )
2. Waktu naik ( $t_r$ )
3. Waktu puncak ( $t_p$ )
4. Waktu turun ( $t_s$ )
5. Maksimum overshoot ( $M_p$ )



Gambar 2.2. Respon Kinerja Sistem

## 3. PERANCANGAN DAN IMPLEMENTASI

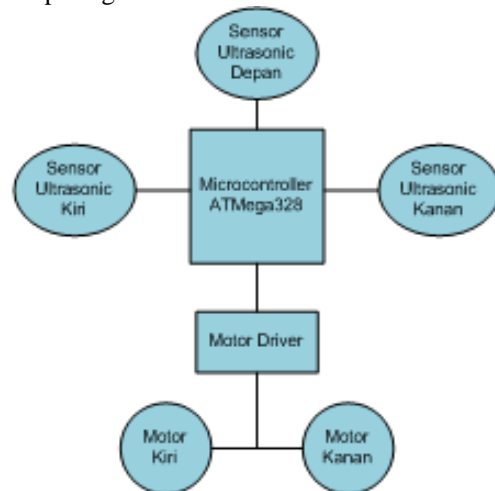
Sistem memiliki input berupa set point posisi robot terhadap dinding. Posisi robot sebenarnya akan dideteksi oleh sensor jarak yaitu sensor ultrasonik. Selisih error yang didapatkan akan diolah oleh didalam mikrokontroler dengan metode kontrol PID. Hasil keluaran kontroler PID akan mengatur PWM pada mikrokontroler yang akan mengatur kecepatan putar motor DC robot dan akan menghasilkan perubahan posisi pada robot.



Gambar 3.1 Blok Diagram Sistem

### 3.1 Perancangan Hardware

Secara umum perancangan hardware dapat dilihat pada gambar 3.2 di bawah.



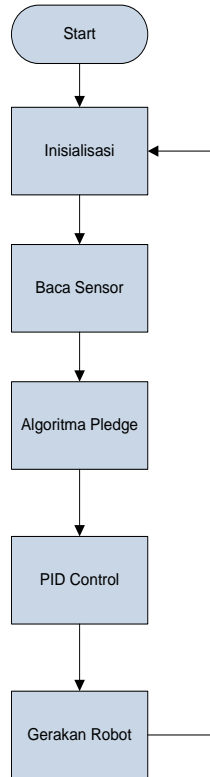
Gambar 3.2 Penempatan Hardware Sistem Robot

Hardware dan alat yang digunakan terdiri dari:

1. Personal Computer (PC)
2. Serial Downloader
3. Sensor Jarak Ultrasonik
4. Mikrokontroler ATmega328p
5. LCD
6. Serial Interface Microcontroller to PC
7. Motor Driver
8. Motor DC

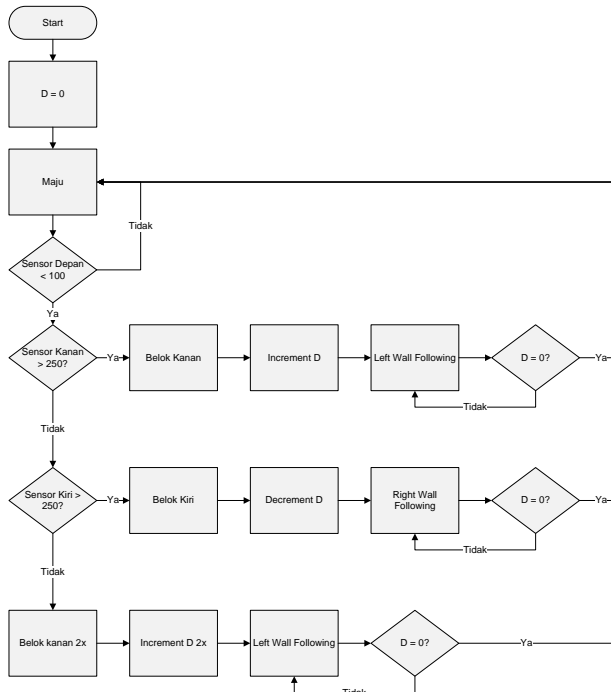
### 3.2 Perancangan Software

Secara umum perancangan software dapat dilihat pada gambar 3.3 di bawah.



Gambar 3.3 Flowchart Perancangan Software

### 3.2.1 Perancangan Algoritma Pledge



Gambar 3.4 Flow Chart Algoritma Pledge

Keputusan yang akan diambil robot pada suatu kondisi tertentu berdasarkan Algoritma *Pledge* adalah sebagai berikut:

1. jika  $d = 0$  & jarak depan  $> 100 \rightarrow$  maju
2. jika  $d = 0$  & jarak depan  $< 100$  & jarak kanan  $> 250 \rightarrow$  belok kanan
3. jika  $d = 0$  & jarak depan  $< 100$  & jarak kanan  $< 250$  & jarak kiri  $> 250 \rightarrow$  belok kiri
4. jika  $d = 0$  & jarak depan  $< 100$  & jarak kanan  $< 250$  & jarak kiri  $< 250 \rightarrow$  belok kanan 2x

Left Wall Following:

5. jika  $d = +X$  & jarak kiri  $> 250 \rightarrow$  belok kiri
6. jika  $d = +X$  & jarak kiri  $< 250$  & jarak depan  $> 100 \rightarrow$  maju
7. jika  $d = +X$  & jarak kiri  $< 250$  & jarak depan  $< 100$  & jarak kanan  $> 250 \rightarrow$  belok kanan
8. jika  $d = +X$  & jarak kiri  $< 250$  & jarak depan  $< 100$  & jarak kanan  $< 250 \rightarrow$  belok kiri 2x

Right Wall Following:

9. jika  $d = -X$  & jarak kanan  $> 250 \rightarrow$  belok kanan
10. jika  $d = -X$  & jarak kanan  $< 250$  & jarak depan  $> 100 \rightarrow$  maju
11. jika  $d = -X$  & jarak kanan  $< 250$  & jarak depan  $< 100$  & jarak kiri  $> 250 \rightarrow$  belok kiri
12. jika  $d = -X$  & jarak kanan  $< 250$  & jarak depan  $< 100$  & jarak kiri  $< 250 \rightarrow$  belok kanan 2x

### 3.2.2 Perancangan PID

Pengendali atau kontroler PID pada robot yang dibuat berfungsi untuk memperhalus gerak robot saat melakukan koreksi jarak terhadap dinding. Pada kontroler ini sensor berfungsi sebagai masukan sistem yang kemudian diolah. Data tersebut akan mengatur keluaran yang berupa nilai PWM untuk mengatur kecepatan motor kiri dan motor kanan.

Secara umum, PID dirumuskan seperti dibawah ini:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad \dots\dots \text{persamaan 3.1}$$

Rumus PID di atas merupakan rumus analog sedangkan untuk mengimplementasikan PID ke dalam mikrokontroler harus diubah ke dalam bentuk diskrit.

$$u_k = K_p e_k + K_i \sum_{n=1}^k e_n + K_d [e_k - e_{k-1}] \quad \dots\dots \text{persamaan 3.2}$$

Untuk mendapatkan nilai parameter  $K_p$ ,  $K_i$  dan  $K_d$  digunakan tuning dengan metode trial & error. Metode tersebut digunakan karena tidak diperlukannya pemodelan sistem secara matematis. Metode tersebut dilakukan dengan mengacu pada sifat kontroler masing-masing.

Tabel 3.1 Sifat masing-masing kontroler PID

Response	Rise Time	Overshoot	Settling Time	Steady State Error
$K_p$	Decrease	Increase	Minor Change	Decrease
$K_i$	Decrease	Increase	Increase	Eliminate
$K_d$	Minor Change	Decrease	Decrease	Minor Change

## 4. PENGUJIAN DAN ANALISA

### 4.1 Pengujian Hardware

#### 4.1.1 Sistem Minimum Atmega328

Dari power supply yang dibuat, tegangan yang masuk ke dalam mikrokontroler bernilai 4.9 volt. Tegangan ini bernilai kurang dari tegangan operasi maksimum yaitu 6 volt. Dengan demikian disimpulkan bahwa power supply dari sistem minimum mikrokontroler yang dibuat berjalan dengan baik.

Hasil pengujian port juga menunjukkan hasil yang memuaskan dimana semua port dapat bekerja dengan baik. Led dapat menyala pada tiap-tiap port.

Hasil pengujian memasukkan program ke dalam chip berjalan dengan baik. Program yang dibuat dengan software BASCOM dapat dimasukkan (download) ke dalam chip mikrokontroler Atmega328 dengan menggunakan software AVROSPIL.

#### 4.1.2 LCD

Dari hasil pengujian yang didapatkan, rangkaian LCD berfungsi dengan baik. LCD dapat menyala, backlight LCD juga menyala dan LCD dapat menampilkan tulisan "Hello World".

#### 4.1.3 Sensor Ultrasonik

Tabel 4.1 Hasil pengujian sensor ultrasonik

Jarak (cm)	timer
3	10
6	20
10	35
20	70
30	105
60	215
90	320
120	430

Dari hasil pengujian ketiga sensor ultrasonik, sensor berfungsi dengan baik. Sensor dapat mendeteksi objek yang berada didepannya

dengan hasil yang tidak jauh berbeda antara ketiga sensor tersebut. Perbedaan yang terjadi dapat diakibatkan oleh keterbatasan pengamat dalam meletakkan jarak benda yang berada di depan sensor dan juga keterbatasan alat dimana tingkat sensitivitas sensor yang berbeda.

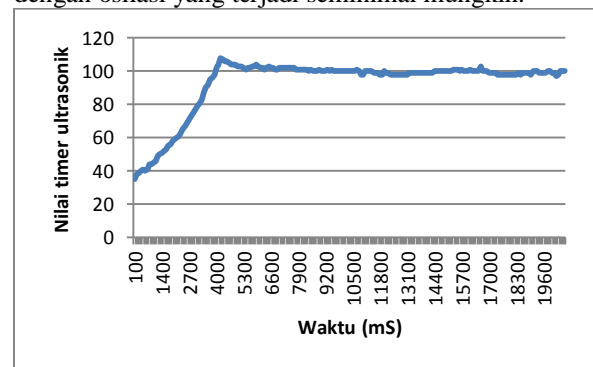
### 4.1.4 Motor Driver

Dari hasil pengujian, dihasilkan motor driver yang dapat bekerja dengan baik dimana motor dapat merubah arah putar sesuai dengan yang diinginkan yaitu searah jarum jam (*clockwise*) dan berlawanan arah jarum jam (*counter clockwise*), dan putaran motor juga dapat dipercepat dan diperlambat sesuai dengan nilai PWM yang diberikan.

## 4.2 Pengujian Software

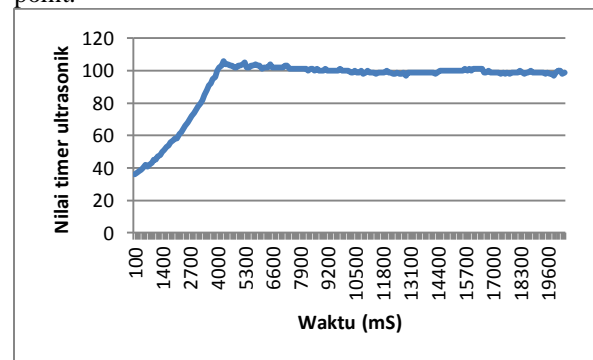
### 4.2.1 Pengujian PID

Dari hasil percobaan *Trial & Error* nilai parameter  $K_p$ , akan digunakan parameter  $K_d$  dengan nilai 70. Hal ini dikarenakan robot memerlukan *rise time* yang cepat dan juga *settle time* yang cepat dengan osilasi yang terjadi seminimal mungkin.



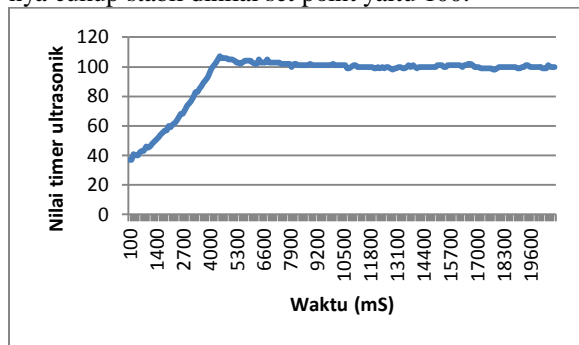
Gambar 4.1 Grafik performansi sistem saat  $K_p = 70$

Dari hasil percobaan *Trial & Error* nilai  $K_d$ , digunakan nilai  $K_d = 30$ . Nilai ini digunakan karena *maximum overshoot* dari sistem berkurang dan *time settle* sistem juga berkurang. Terlihat dari grafik performansi sistemnya juga lebih stabil ke nilai set point.



Gambar 4.2 Grafik performansi sistem saat  $K_p = 70$  dan  $K_d = 30$

Dari hasil percobaan *Trial & Error* nilai  $K_i$ , digunakan nilai  $K_i = 0.05$ . Hasil ini digunakan karena terlihat dari grafik performansi sistemnya, *settle time*-nya cukup stabil dinilai set point yaitu 100.

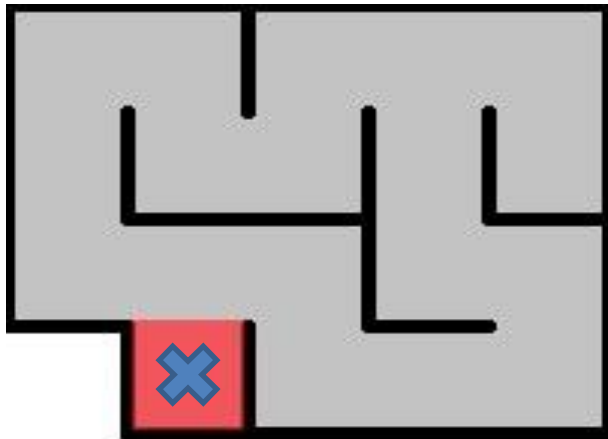


Gambar 4.3 Grafik performansi sistem saat  $K_p = 70$ ,  $K_d = 30$  dan  $K_i = 0.05$

dengan nilai parameter  $K_p=70$ ,  $K_d=30$ ,  $K_i=0.05$  menghasilkan *maximum overshoot* senilai 107 atau 7 %, *rise time* bernilai 4100ms, *delay time* 2800ms, *peak time* 4500ms dan *settle time* bernilai 7600ms.

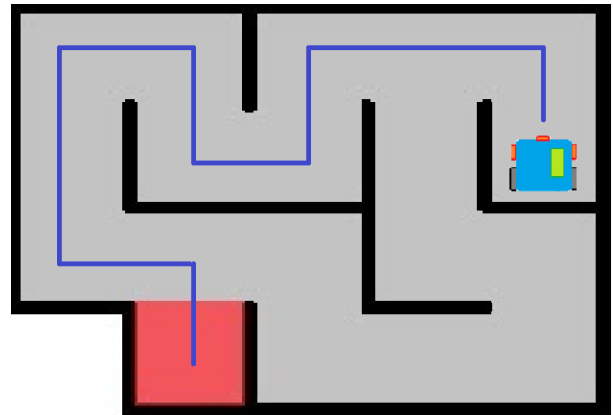
#### 4.2.2 Pengujian Program Algoritma *Pledge*

Labirin yang akan digunakan dalam pengujian program algoritma *Pledge* adalah sebagai berikut.

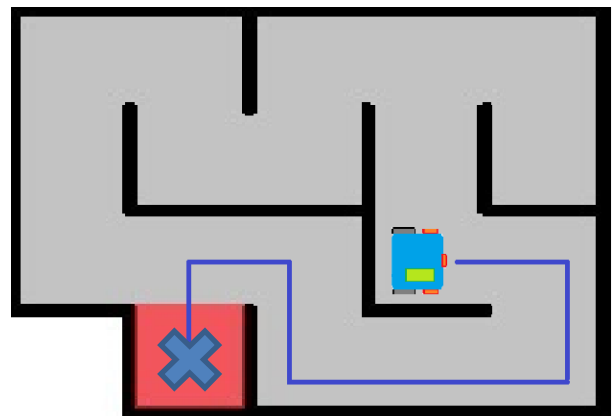


Gambar 4.4 Labirin yang digunakan dalam pengujian robot

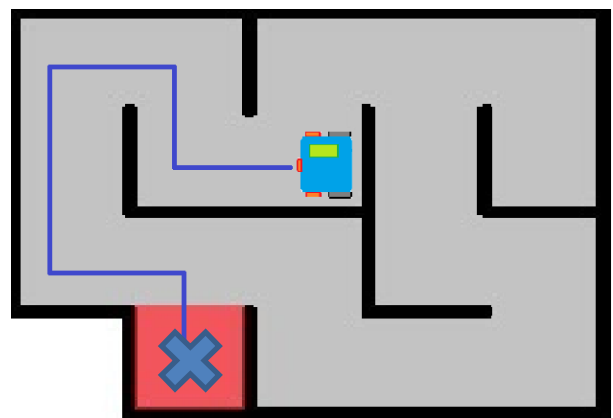
Dari hasil pengujian yang didapatkan, pergerakan robot pada titik awal robot yang berbeda-beda dapat menemukan jalan keluar dari labirin tersebut. Hasil dari masing-masing pergerakan robot dapat dilihat sebagai berikut:



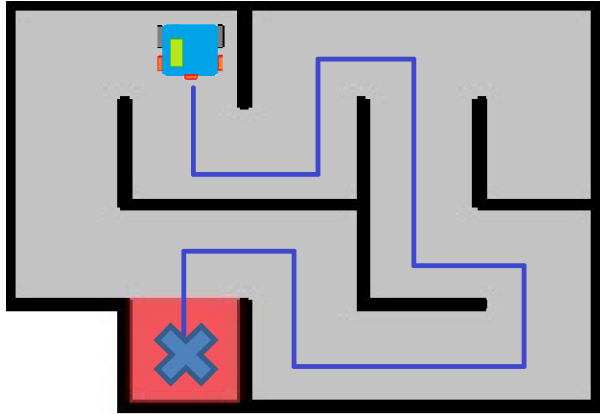
Gambar 4.5 Pergerakan robot menuju jalan keluar pada titik awal 1



Gambar 4.6 Pergerakan robot menuju jalan keluar pada titik awal 2



Gambar 4.7 Pergerakan robot menuju jalan keluar pada titik awal 3



Gambar 4.8 Pergerakan robot menuju jalan keluar pada titik awal 4

Dari hasil keseluruhan yang didapatkan, dapat disimpulkan bahwa algoritma *Pledge* yang diterapkan ke dalam sistem robot bekerja dengan baik untuk dapat menyelesaikan labirin yang diberikan. Dengan penambahan kontroler PID robot dapat berjalan dengan baik pada lintasannya. Waktu yang dibutuhkan robot untuk menyelesaikan labirin tersebut bervariasi sesuai dengan posisi awal robot.

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dari hasil pengujian dan analisis yang telah dilakukan pada perancangan sistem robot dengan menggunakan algoritma *Pledge* sebagai algoritma pencarian jalan keluar labirin, dapat diambil kesimpulan sebagai berikut:

1. Performansi perangkat keras (*hardware*) robot terintegrasi yang terdiri dari sistem minimum mikrokontroler Atmega 328p, tiga sensor ultrasonik, LCD 16x2, driver motor L298N, dan 2 buah motor DC 24V bekerja dengan baik.
2. Performansi dari sistem robot lapangan yang menggunakan algoritma PID sebagai pengontrolnya yang meliputi *rise time* ( $t_r$ ), *delay time* ( $t_d$ ), *peak time* ( $t_p$ ), *settling time* ( $t_s$ ), dan *maximum overshoot* ( $M_p$ ) dipengaruhi oleh parameter konstanta P ( $K_p$ ), konstanta I ( $K_i$ ) dan konstanta D ( $K_d$ ).
3. Dari percobaan yang dilakukan, didapatkan nilai konstanta dari ketiga parameter tersebut agar mencapai performansi yang diinginkan yaitu sebesar  $K_p = 70$ ,  $K_d = 30$  dan  $K_i = 0.05$  dengan performansi *Maximum overshoot* ( $M_p$ ) sebesar 7%, *rise time* ( $t_r$ ) sebesar 4100ms, *delay time* ( $t_d$ ) sebesar 2800ms, *peak time* ( $t_p$ ) sebesar 4500ms, *settling time* ( $t_s$ ) sebesar 7600ms.

4. Pengimplementasian algoritma *Pledge* dan kontroler PID ke dalam bahasa Basic berhasil dilakukan. Ini terbukti dari sifat pergerakan robot pada saat melintasi labirin yang telah disediakan, dimana robot berjalan tanpa menabrak dinding labirin dan robot mampu menemukan jalan keluar dari labirin dari titik awal yang berbeda-beda.
5. Dari percobaan yang dilakukan pada labirin yang tersedia, robot yang menggunakan algoritma *Pledge* dapat menemukan jalan keluar dari labirin dengan tingkat keberhasilan 100%.

### 5.2 Saran

Pengembangan yang dapat dilakukan untuk menyempurnakan tugas akhir ini adalah:

1. Penggunaan sensor tambahan agar robot dapat berjalan lurus pada lintasannya saat tidak melakukan wall following.
2. Menggunakan Algoritma yang lebih baik dibandingkan dengan Algoritma *Pledge* dalam hal pencarian jalan keluar labirin dimana dapat mencari jalan keluar dengan jalur yang lebih pendek dibanding algoritma *Pledge*.
3. Memodelkan sistem ke dalam fungsi matematika sehingga dapat dicari performansi yang diinginkan dengan lebih akurat.

### DAFTAR PUSTAKA

- [1] Ogata, Katshuhiko. 1997. *Teknik Kontrol Otomatis*. Jakarta : Penerbit Erlangga,
- [2] Budiharto, Widodo. 2009. *Membuat Sendiri Robot Cerdas*. Jakarta : Elex Media Komputindo.
- [3] Budiharto, Widodo. 2010. *Robotika - teori dan implementasi*. Yogyakarta : Penerbit ANDI.
- [4] Primanio. 2009. *Pencarian Jalan Keluar Labirin Dengan Metode Wall Follower*. Bandung : Makalah Institut Teknologi Bandung.
- [5] Wicaksono, Doni. 2011. *Implementasi Pengontrolan Kecepatan Motor DC pada Robot Beroda Menggunakan Logika Fuzzy Berdasarkan Pembacaan Sensor Ultrasonik*. Bandung : Tugas Akhir Institut Teknologi Telkom.
- [6] Herdianto, Candra. 2012. *Desain dan Implementasi Sistem Navigasi Robot Beroda Menggunakan Algoritma Wall Following Berbasis PID*. Bandung : Tugas Akhir Institut Teknologi Telkom.