

Makalah Seminar Tugas Akhir

PENGENDALIAN ROBOT MOBIL PENCARI TARGET DENGAN KEMAMPUAN MENGHINDARI RINTANGAN

R. Kurniawan I. L.^[1], Iwan Setiawan, ST, MT^[2], Darjat, ST, MT^[2]
Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia

ABSTRAK

Kebutuhan manusia akan kemudahan yang semakin meningkat menuntut adanya otomatisasi di segala bidang. Teknologi robot mobil sebagai alat bantu manusia terus berkembang sekarang ini untuk menjawab tantangan itu. Dalam perancangan robot mobil terutama yang memiliki misi "menuju sasaran" dengan tipe penggerak diferensial biasanya hanya terdapat satu kontroler close loop untuk mengendalikan posisi robot. Sedangkan kendali kecepatan ke roda kanan dan kiri untuk mengatur pergerakan robot dilakukan secara open loop atau dengan kata lain kecepatan roda aktual tidak di perhatikan. Hal ini menyebabkan kelebihan-kelebihan dari metode kontrol yang digunakan tidak akan berpengaruh banyak dalam pengendalian robot, sebab penggerak robot tidak selalu bekerja sesuai harapan. Tujuan pembuatan tugas akhir ini adalah membuat membuat sistem kendali robot yang mampu bergerak menuju sasaran yang diinginkan pengguna dengan dua tingkat pengontrolan, yaitu kendali posisi dan kendali kecepatan.

Objek yang akan dikendalikan adalah robot jenis diferensial, yang mempunyai dua motor DC sebagai penggerak roda kanan dan kiri. Pengendali P digunakan sebagai kendali posisi robot. Sedangkan kecepatan motor dikendalikan dengan metode kontrol PI. Digunakan 3 buah sensor ultrasonik untuk mendeteksi adanya halangan pada lintasan robot.

Dari pengujian, didapatkan bahwa kendali close loop kecepatan aktual robot dapat meningkatkan unjuk kerja robot mobil dalam mencapai sasaran. Robot berhenti pada jarak 3cm-5.8cm dari sasaran. Secara keseluruhan robot mobil mampu mencapai sasaran dengan lebih baik pada jarak relatif dekat.

Kata-kunci: robot mobil, penggerak diferensial, proporsional, PI

I PENDAHULUAN

1.1 Latar Belakang

Teknologi robot mobil sebagai alat bantu manusia terus berkembang sekarang ini untuk menjawab tantangan itu. Mulai dari tujuan hiburan, robot mobil cerdas, alat pengangkut barang, hingga misi luar angkasa telah menggunakan teknologi robot mobil sebagai alat bantu manusia. Dengan melihat peran robot mobil yang semakin besar, dituntut adanya metode-metode baru untuk penyempurnaan teknologi robot mobil ini.

Pada robot mobil, terutama yang memiliki misi "menuju sasaran" dengan tipe penggerak diferensial biasanya hanya terdapat satu kontroler *close loop* untuk mengendalikan posisi robot. Sedangkan kendali kecepatan ke roda kanan dan kiri untuk mengatur pergerakan robot dilakukan secara *open loop* atau dengan kata lain kecepatan roda aktual tidak di perhatikan. Mengingat banyak sekali hal yang mempengaruhi kecepatan putar roda seperti tidak identiknya kinerja kedua roda, berat robot dan lintasan robot yang berupa tanjakan atau turunan, membuat metode pengendalian kecepatan secara *open loop* ini sangat merugikan.

Dalam tugas akhir ini akan dibuat sebuah robot mobil dengan penggerak diferensial yang

mempunyai kemampuan untuk menuju target posisi yang diinginkan. Pengendali proporsional digunakan untuk mengontrol pergerakan robot. Selain itu robot juga akan dilengkapi dengan metode kontrol PI untuk mengatur kecepatan roda aktual robot.

1.2 Tujuan

Tujuan pembuatan tugas akhir ini adalah membuat robot mobil menggunakan sistem kendali posisi Proporsional dan sistem kendali kecepatan Proporsional-Integral dengan kemampuan menghindari rintangan untuk mencapai posisi tujuan yang diinginkan pengguna.

1.3 Pembatasan Masalah

Dalam pembuatan tugas akhir ini penulis membatasi permasalahan sebagai berikut :

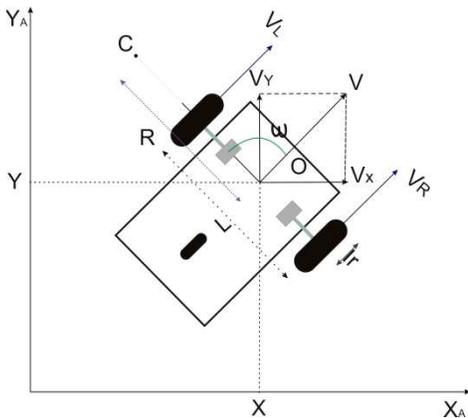
1. Mikrokontroler yang digunakan adalah ATmega32.
2. Sensor jarak yang digunakan dari jenis ultrasonik (PINGTM)
3. Medan berupa bidang datar dua dimensi.
4. Robot mobil yang dibuat menggunakan penggerak diferensial.
5. Kendali posisi dengan metode kontrol Proporsional.

6. Kendali kecepatan dengan metode kontrol Proporsional-Integral.
7. Tidak membahas mekanik robot.

II DASAR TEORI

2.1 Model Kinematika Robot mobil Penggerak Differensial

Robot mobil tipe penggerak diferensial memiliki 2 buah roda penggerak yang terpisah (kanan dan kiri). Kedua roda ini digerakkan oleh motor DC yang ditempatkan pada satu sumbu secara terpisah. Sehingga kedua roda ini berfungsi sebagai penggerak sekaligus sebagai kemudi robot mobil. Sebagai penyeimbang umumnya robot mobil ini dilengkapi juga dengan satu atau dua buah roda castor yang ditempatkan dibagian belakang robot mobil tersebut



Gambar 2.1 Posisi dan Orientasi robot mobil dalam Sistem Koordinat Cartesian.

Secara matematis kecepatan linier dan kecepatan angular robot mobil dirumuskan sesuai persamaan berikut :

$$\begin{bmatrix} V(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} V_R(t) \\ V_L(t) \end{bmatrix}$$

Persamaan diatas pada dasarnya memperlihatkan relasi kinematika langsung antara kecepatan linier roda-roda robot terhadap kecepatan linier dan angular robotnya, sedangkan Persamaan berikut memperlihatkan relasi sebaliknya (Kinematika Balik).

$$\begin{bmatrix} V_R(t) \\ V_L(t) \end{bmatrix} = \begin{bmatrix} 1 & \frac{L}{2} \\ 1 & -\frac{L}{2} \end{bmatrix} \begin{bmatrix} V(t) \\ \omega(t) \end{bmatrix}$$

2.2 Sistem Kendali PID

Pengendali PID adalah suatu sistem pengendali yang merupakan gabungan antara pengendali proporsional, integral, dan turunan (*derivative*). Dalam waktu kontinyu, sinyal keluaran pengendali PID dapat dirumuskan sebagai berikut.

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right)$$

atau

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

dengan

$u(t)$ = sinyal keluaran pengendali PID

K_p = konstanta proporsional

T_i = waktu integral

T_d = waktu turunan

K_i = konstanta integral

K_d = konstanta turunan

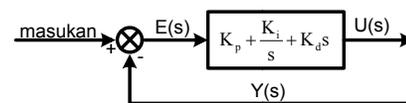
$e(t)$ = sinyal kesalahan

($e(t) = referensi - output$)

Jadi, fungsi alih pengendali PID (dalam domain s) dapat dinyatakan sebagai berikut.

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s$$

Diagram blok pengendali PID dapat dilihat pada Gambar 2.2.



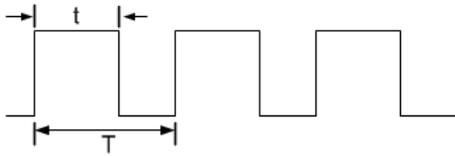
Gambar 2.2 Diagram blok pengendali PID.

2.3 Pulse Width Modulation (PWM)

PWM adalah salah satu cara mengatur kecepatan motor dc dengan menggunakan sumber tegangan dc yang tetap, yaitu dengan men-switch secara cepat antara kondisi *ON* dan *OFF* pada frekuensi tertentu. Untuk mendapatkan perputaran yang halus biasanya frekuensi yang dipakai adalah 1KHz atau lebih. Kondisi *ON* (logika high) sebagai lebar pulsa yang

akan diatur adalah t , dan periode sinyal adalah T , dimana:

$$T = \frac{1}{f}$$



Gambar 2.3 Sinyal PWM.

Rasio perbandingan antara kondisi ON dengan periode sinyal disebut sebagai duty cycle, dinyatakan dengan:

$$\text{duty cycle} = \frac{t}{T} \times 100\%$$

Besar presentase duty cycle dengan kecepatan motor adalah berbanding lurus, dimana semakin besar duty cycle maka arus rata-rata yang melalui motor juga semakin besar sehingga motor akan semakin cepat. Demikian pula semakin kecil duty cycle maka arus rata-rata yang melalui motor juga kecil sehingga motor akan bergerak lambat.

2.4 Odometri

Teknik klasik bagi robot beroda dalam menghitung posisinya adalah dengan menjejak lokasi robot tersebut melalui sebuah pengukuran secara kontinyu dari rotasi roda robot, metode ini sering disebut dengan Odometri. Odometri merupakan metode navigasi yang paling sering digunakan untuk menentukan posisi robot mobil.

Perhitungan jarak tempuh robot dengan tingkat ketelitian per 1 cm digunakan konstanta "banyaknya lubang yang terdeteksi untuk jarak tempuh 1 cm," yang didapat dari perhitungan berikut.

$$RC = \frac{\text{Jumlah Lubang Encoder}}{\text{Keliling Lingkaran}}$$

Untuk robot yang dikendalikan secara *differential*, lokasi robot selalu diperbaharui secara tetap menggunakan rumus di bawah ini.

$$\text{Jarak}_R = (\text{Jumlah Pulsa Roda Kanan}) / RC$$

$$\text{Jarak}_L = (\text{Jumlah Pulsa Roda Kiri}) / LC$$

$$\text{Jarak} = (\text{Jarak}_R + \text{Jarak}_L) / 2$$

$$\theta = \theta_0 + \frac{(\text{Jarak}_R - \text{Jarak}_L)}{L}$$

$$X = X_0 + \text{Jarak} * \cos(\theta)$$

$$Y = Y_0 + \text{Jarak} * \sin(\theta)$$

Keterangan :

Jarak_R = Jarak tempuh roda kanan robot

Jarak_L = Jarak tempuh roda kiri robot

RC = Konstanta banyaknya lubang yang terdeteksi untuk jarak tempuh roda kanan sejauh 1 cm

LC = Konstanta banyaknya lubang yang terdeteksi untuk jarak tempuh roda kiri sejauh 1 cm

L = Jarak antara roda kanan dan roda kiri

Jarak = Jarak yang ditempuh oleh robot keseluruhan

θ = Sudut atau orientasi yang dibentuk oleh robot dalam radian

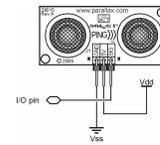
X = Posisi robot dalam absis

Y = Posisi robot dalam ordinat

Dengan membangkitkan fungsi-fungsi di atas secara kontinyu maka akan didapatkan nilai X , Y dan θ setiap saat.

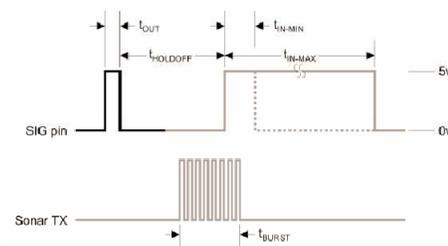
2.5 Sensor Jarak Ultrasonik PING)))™

Sensor yang digunakan adalah "PING)))™ Ultrasonic Range Finder", buatan Parallax. Sensor jarak ditunjukkan pada gambar 2.3.



Gambar 2.4 Sensor PING)))™ Ultrasonic Range Finder.

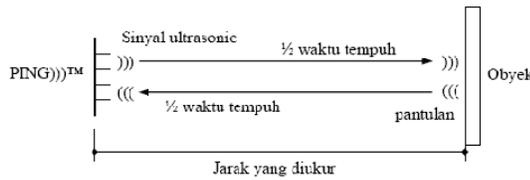
Diagram pewaktuan sensor ditunjukkan pada gambar 2.5.



Gambar 2.5 Diagram pewaktuan sensor Ping)))™.

Agar sensor ini dapat digunakan untuk mengukur jarak dibutuhkan sebuah mikrokontroler untuk mengukur waktu tempuh sinyal ultrasonik dari sensor, memantul pada benda yang akan

diukur, dan diterima kembali oleh sensor. Ilustrasi cara kerja sensor ditunjukkan pada Gambar 2.6.



Gambar 2.6 Ilustrasi cara kerja Ping)))TM.

Besarnya jarak yang diukur dapat dihitung dengan menggunakan persamaan :

$$s = \frac{v \cdot t}{2}$$

dengan: s = jarak yang diukur (meter)
 v = kecepatan suara (344 m/detik)
 t = waktu tempuh (detik)

III PERANCANGAN SISTEM

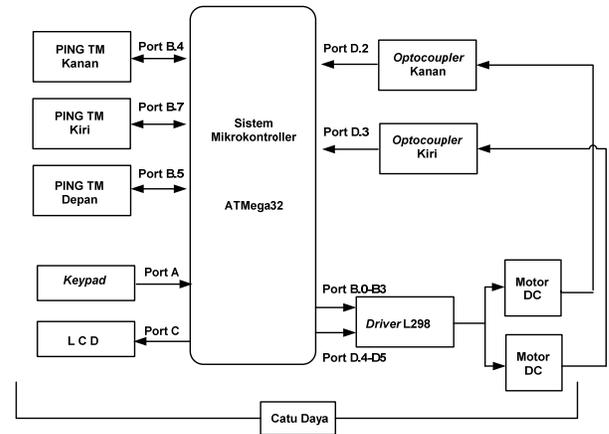
Pada perancangan “ Robot Mobil Pencari Target Dengan Kemampuan Menghindari Rintang ” akan dibagi menjadi 2 yaitu perancangan perangkat keras (*Hardware*) dan perangkat lunak (*Software*).

3.1 Perancangan Perangkat Keras (*Hardware*)

Perangkat keras dari sistem yang akan dibangun meliputi sistem minimum mikrokontroler ATmega32, sensor PING)))TM, Keypad, LCD, driver motor dc, dan optocoupler.

3.1.1 Sistem Mikrokontroler ATmega32

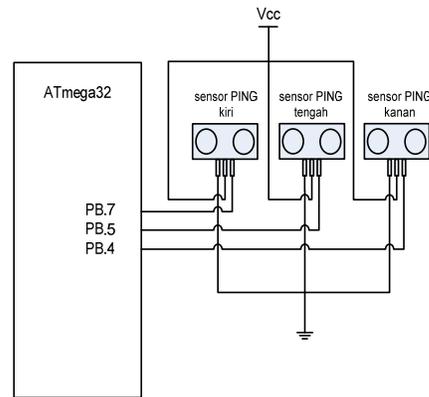
Untuk melakukan fungsi-fungsinya tersebut di atas, maka dilakukan pengalokasian penggunaan port yang ada pada mikrokontroler ATmega32. PORTA digunakan sebagai masukan keypad. PORTB digunakan untuk masukan dari sensor PING)))TM (PORTB.4, PORTB.5 dan PORTB.7), dan juga sebagai masukan pengatur arah putaran motor bagi driver motor (PORTB.0 sampai PORTB.3) . PORTC berfungsi sebagai keluaran pengatur tampilan LCD. PORTD sebagian difungsikan keluaran PWM yang digunakan untuk mengatur pemberian tegangan ke driver motor DC (PORTD.4 dan PORTD.5) , PORTD.2 dan PORTD.3 digunakan untuk menerima informasi dari optocoupler. Untuk lebih jelasnya, konfigurasi penempatan masukan dan keluaran pada Mikrokontroler ATmega32 dapat dilihat pada Gambar 3.1



Gambar 3.1 Blok diagram sistem.

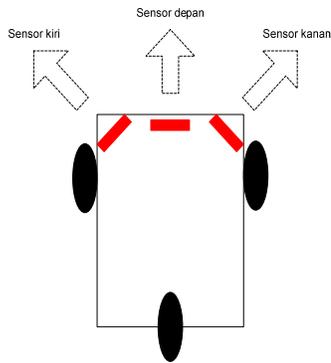
3.1.2 Sensor PING)))TM

Alokasi port untuk sensor PING)))TM dapat dilihat pada Gambar 3.2.



Gambar 3.2 Alokasi port alamat sensor PING)))TM.

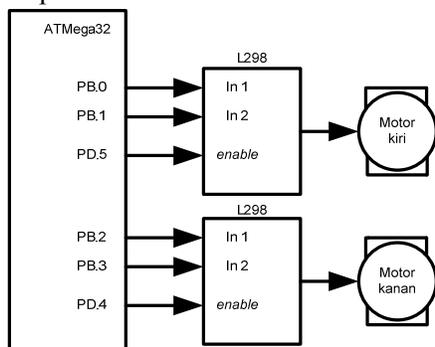
Berdasarkan Gambar 3.5 dapat diketahui bahwa sensor PING)))TM memiliki 3 kaki. Kaki SIG I/O dihubungkan dengan PINB mikrokontroler, untuk kaki GND dan VCC masing-masing dihubungkan dengan ground dan sumber tegangan. Pada perancangan digunakan tiga buah sensor PING)))TM yang terletak di kanan, depan dan kiri robot. Sensor pada kiri dan kanan robot terletak pada bagian depan robot dengan posisi menyerong. Sensor diletakan pada robot dengan ketinggian 20 cm dari permukaan tanah.



Gambar 3.3 Peletakkan sensor PING)))TM pada robot.

3.1.3 Perancangan Driver L298

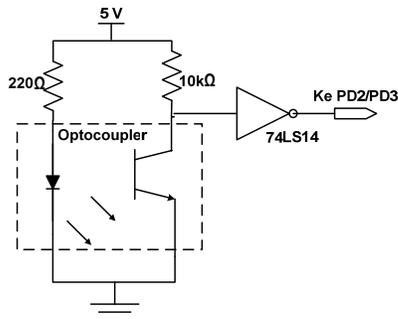
Alokasi *port driver* L298 untuk menggerakkan motor dc diperlihatkan pada Gambar 3.4. Masukan dari rangkaian driver adalah sinyal PWM yang dibangkitkan pada PD.4 dan PD.5 mikrokontroler ATmega32. Sinyal PWM akan mengatur tegangan masukan ke motor dengan mengatur lebar pulsanya. Arah putaran motor diatur oleh PB1 hingga PB3.



Gambar 3.4 Alokasi port untuk driver L298.

3.1.4 Perancangan Optocoupler

Gambar 3.5 berikut menunjukkan diagram rangkaian *optocoupler (encoder)*.



Gambar 3.5 Rangkaian *optocoupler*.

Prinsip kerja *optocoupler* adalah mengubah sinyal *on/off* dari sinar yang masuk ke foto-transistor menjadi bentuk sinyal digital 0V/5V.

Pemanfaatannya dengan cara memasang piringan berlubang di antara foto-dioda dan foto-transistor yang terhubung ke poros motor, sehingga dapat digunakan untuk mengukur kecepatan motor berdasarkan frekuensi digital yang dihasilkan.

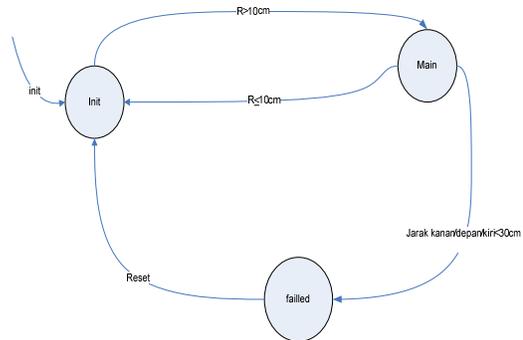
Keluaran *optocoupler* digunakan sebagai intrupsi eksternal pada ATmega32 yang kemudian diolah sebagai umpan balik kecepatan dan posisi.

3.2 Perancangan Software

Perangkat lunak dari sistem yang akan dibangun meliputi sistem kendali utama, sistem kendali, pengaturan pergerakan motor, sistem *update* jarak, penggunaan *keypad*, pembacaan sensor PING)))TM, pembacaan *encoder* dan tampilan LCD.

3.1.5 Perancangan Program Utama

Program dimulai dengan melakukan inialisasi semua variabel-variabel utama dari proses pengendalian dan mikrokontroler.

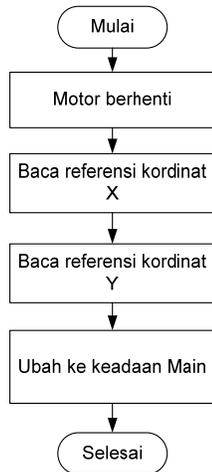


Gambar 3.5 Diagram keadaan program utama.

Pada program utama terdapat beberapa keadaan dalam logika robot. Setiap saat robot hanya berada pada satu keadaan. Dalam program utama terdapat keadaan *init*, keadaan *Main* dan keadaan *Failed*. Selain itu juga terdapat beberapa tambahan untuk menunjang pemilihan keadaan oleh robot, seperti keadaan *reset* dan pembacaan sensor PING)))TM

3.1.5.1 Keadaan *Init*

Keadaan *init* adalah keadaan dimana robot pertamakali dihidupkan, setelah robot menyelesaikan tugas dan ketika robot di reset setelah gagal menemukan sasaran.

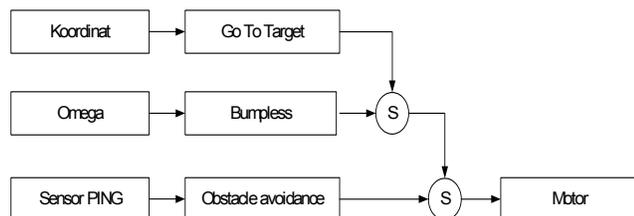


Gambar 3.6 Diagram alir keadaan Init.

Dalam keadaan *Init* kordinat acuan robot didapatkan melalui masukan *keypadd*.

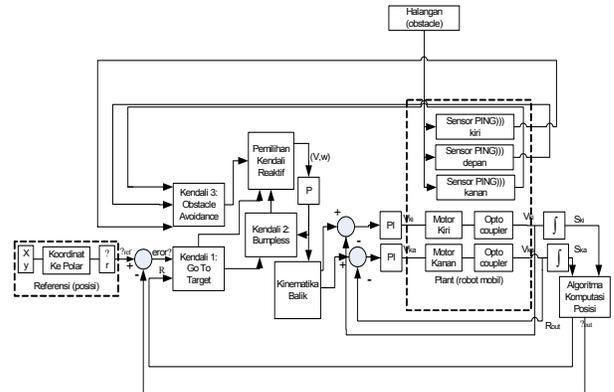
3.1.5.2 Keadaan *Main*

Keadaan *Main* adalah keadaan utama yang ada pada robot. Setelah robot menerima koordinat sasaran dari keadaan *Init* diatas, kemudian pada keadaan *Main* robot akan bergerak menuju sasaran yang diinginkan. Terdapat beberapa kendali reaktif dalam keadaan ini, yaitu *Go To Target*, *Bumpless* dan *Obstacle Avoidance*. Kendali *Go To Target* adalah kendali utama dari robot. Kendali *Bumpless* adalah kendali transisi antara kendali *Go To Target* dengan kendali *Obstacle Avoidance*. Kendali ini digunakan untuk meminimalisir hentakan yang terjadi ketika terjadi perpindahan kendali. Kendali yang terakhir adalah kendali *Obstacle Avoidance*, kendali ini digunakan oleh robot untuk menghindari rintangan yang mungkin ada saat bergerak menuju sasaran. Untuk pengaturannya kendali tersebut diatur secara reaktif dengan arsitektur *subsumption*. Untuk diagramnya dapat dilihat dalam diagram pada Gambar 3.7.



Gambar 3.7 Blok diagram arsitektur *subsumption*.

Dalam pengaturannya terdapat beberapa pengendali untuk mengatur pergerakan robot. Secara lengkap pengendaliannya dapat dilihat dalam blok diagram pada Gambar 3.8.



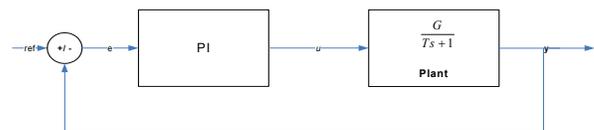
Gambar 3.8 Blok diagram pengendalian pada keadaan *Main*.

3.1.5.3 Keadaan *Failed*

Pada keadaan *failed* robot akan dihentikan dan menunggu tombol reset ditekan untuk kembali ke keadaan *init*.

3.1.6 Pengendalian Kecepatan Aktual Robot

Pada perancangan program utama terdapat pengendalian posisi robot yaitu dengan menggunakan kendali Proporsional. Contoh dari aksi kendali proporsional adalah pada pengaturan omega (kecepatan angular). Selanjutnya pada program utama dihasilkan nilai kecepatan referensi untuk roda kiri dan roda kanan. Kelebihan dari sistem robot ini adalah kecepatan aktual dari tiap roda diumpan balik kembali dan di kendalikan secara *close loop*. Pengendalian ini berbasis kendali PI (Proporsional-Integral). Karena pengendalian dalam kawasan diskrit maka diperlukan penetapan waktu cuplik. Kendali kecepatan aktual robot akan dieksekusi setiap 0.05s sekali.



Gambar 3.9 Diagram blok kontrol kecepatan aktual.

3.1.7 Update Posisi

Update posisi berfungsi untuk mengetahui letak robot setiap saat dalam sistem koordinat serta posisi relatifnya terhadap target. Besarnya waktu cuplik diatur lewat interupsi *timer 0*.

Adapun persamaan yang digunakan adalah sebagai berikut:

$$Jarak_R = (\text{Jumlah Pulsa Roda Kanan}) / RC$$

$$Jarak_L = (\text{Jumlah Pulsa Roda Kiri}) / LC$$

$$Jarak = (Jarak_R + Jarak_L) / 2$$

$$\theta = \theta_0 + \frac{(Jarak_R - Jarak_L)}{L}$$

$$X = X_0 + Jarak * \cos(\theta)$$

$$Y = Y_0 + Jarak * \sin(\theta)$$

Keterangan :

$Jarak_R$ = Jarak tempuh roda kanan robot

$Jarak_L$ = Jarak tempuh roda kiri robot

RC = Konstanta banyaknya lubang yang terdeteksi untuk jarak tempuh roda kanan sejauh 1 cm

LC = Konstanta banyaknya lubang yang terdeteksi untuk jarak tempuh roda kiri sejauh 1 cm

L = Jarak antara roda kanan dan roda kiri

$Jarak$ = Jarak yang ditempuh oleh robot keseluruhan

θ = Sudut atau orientasi yang dibentuk oleh robot dalam radian

X = Posisi robot dalam absis

Y = Posisi robot dalam ordinat

IV PENGUJIAN DAN ANALISA

Pengujian yang dilakukan dalam perancangan Tugas Akhir ini meliputi pengujian unit masukan, pengujian unit keluaran, pengujian kendali dan pengujian tingkah laku robot.

4.1 Pengujian Unit Masukan

Pengujian unit masukan meliputi pengujian sensor-sensor pada robot yang terdiri atas *encoder*, sensor PING™.

4.1.1 Pengujian *Optocoupler*

Pengujian *optocoupler* merupakan pengujian terhadap jumlah intrupsi yang terjadi karena perubahan keadaan keluaran *optocoupler (falling edge)* pada mikrokontroler tiap putaran roda. Pengujian dilakukan dengan memutar roda robot secara manual satu putaran dan membaca keluaran jumlah interupsi yang telah terjadi. Hasil pengujian dapat dilihat pada Tabel 4.1 berikut:

Tabel 4.1 Hasil pengujian *optocoupler*

No Percobaan	<i>Optocoupler</i> Kanan (Interupsi)	<i>Optocoupler</i> Kiri (Interupsi)
1	1270	1355
2	1273	1352
3	1272	1353
4	1275	1354
5	1273	1354
6	1274	1354
7	1275	1353
8	1274	1353
9	1273	1355
10	1276	1355
Rata-rata	1273.5	1353.8

Pengujian pembacaan *optocoupler* menunjukkan perbedaan yang cukup mencolok antara jumlah interupsi *optocoupler* kiri dan *optocoupler* kanan. Hal ini dapat terjadi karena gigi-gigi yang terpasang pada kedua roda tidak identik, selain itu pemasangan *optocoupler* pada motor kiri dan kanan yang kurang tepat mengakibatkan hilangnya beberapa informasi dari pembacaan *optocoupler*. Kekurang sempurnan dari perancangan *optocoupler* telah dikompensasi dengan penyesuaian pada sisi perangkat lunak.

4.1.2 Pengujian Sensor PING™

Pengujian terhadap sensor PING™ dilakukan dengan mengukur jarak suatu benda terukur yang diletakkan di hadapan sensor. Pembacaan sensor dibatasi pada jarak 10 cm – 100 cm. Hasil pengujian dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil pengukuran jarak dengan menggunakan PING™

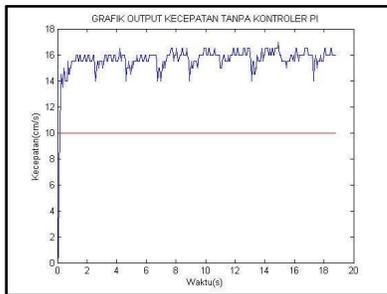
No.	Jarak Uji (cm)	Jarak Terdeteksi PING™		
		PING™ kiri (cm)	PING™ tengah (cm)	PING™ kanan (cm)
1	10	9.1	10.8	10.3
2	20	19.6	20.1	20.4
3	30	29.9	30.9	30.2
4	40	40	41.4	40.7
5	50	50.5	51.2	51
6	60	61	61.5	61.4
7	70	70.5	72	71.3
8	80	79.9	82.3	81.1
9	90	91.8	92.3	92.3
10	100	102.1	103.7	100.9

Dari Tabel 4.2, terlihat bahwa sensor PING™ tidak sesuai terhadap jarak sesungguhnya. Hal ini dapat disebabkan karena pengukuran atau adanya pembulatan dalam pemrograman.

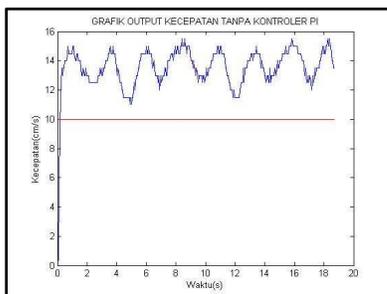
4.2 Pengujian Unit Keluaran

4.2.1 Pengujian Motor DC

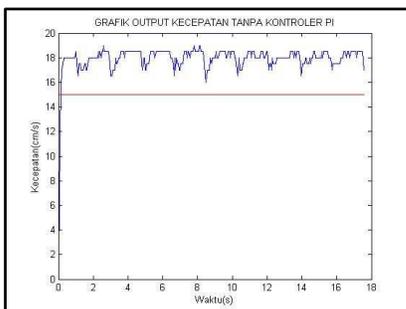
Pengujian motor DC yang dilakukan adalah perbandingan tanggapan kecepatan motor kiri dan kanan dengan masukan nilai PWM yang sama. Pengujian dilakukan dengan memberikan nilai masukan PWM yang sama pada motor kiri dan kanan sebesar 165 dan 195. Grafik tanggapan motor dapat dilihat pada Gambar 4.1 hingga Gambar 4.4.



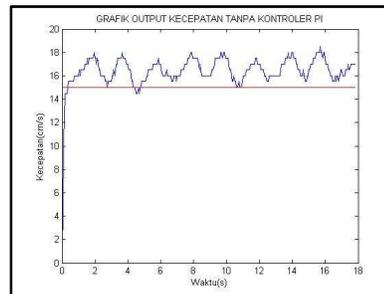
Gambar 4.1 Sinyal keluaran motor DC kanan tanpa beban dengan nilai PWM =165.



Gambar 4.2 Sinyal keluaran motor DC kiri tanpa beban dengan nilai PWM =165.



Gambar 4.3 Sinyal keluaran motor DC kanan tanpa beban dengan nilai PWM =195.



Gambar 4.4 Sinyal keluaran motor DC kiri tanpa beban dengan nilai PWM =195.

Keluaran motor DC tampak berisilasi pada saat pengujian, hal ini dikarenakan ketidaksempurnaan dalam pembuatan mekanik robot, pada beberapa sudut putar motor DC terdamp akibat gesekan robot sehingga kecepatannya turun, akan tetapi pada beberapa sudut putar motor DC tidak terdamp dan kecepatannya meningkat kembali. Selain itu osilasi juga dimungkinkan terjadi karena kesalahan pembacaan pulsa oleh *optocoupler*, ada informasi yang hilang pada beberapa titik.

Secara keseluruhan dari hasil pengujian dapat dilihat bahwa tanggapan kedua motor tidak identik. Tetapi motor memberikan tanggapan kecepatan yang sebanding dengan nilai tegangan masukannya.

4.2 Pengujian Kendali Robot

Pengujian kendali robot meliputi pengujian kendali PI (Proporsional Integral) kecepatan aktual roda, pengendalian Proporsional terhadap sudut hadap robot dan pengujian posisi koordinat akhir robot.

4.2.1 Pengujian Kendali Kecepatan dengan Kendali PI (Proporsional Integral)

Pengujian kendali kecepatan dilakukan dengan memberikan *acuan* tertentu pada roda kanan dan roda kiri. Pengujian kemudian dibandingkan dengan tanggapan kecepatan roda secara *open loop*. Pengujian dilakukan empat kali yaitu uji kecepatan naik arah kedepan, uji kecepatan turun arah kedepan, uji kecepatan naik arah kebelakang dan uji kecepatan turun arah kebelakang, dimana sebagai acuan arah adalah arah gerak robot bila mengikuti kecepatan roda.

Tabel 4.3 Hasil uji kendali PI

No.	Pengujian	Waktu mencapai acuan (s)		Galat Tunak (cm/s)	
		kanan	kiri	kanan	kiri
1	Ref naik ke 15cm/s	1.5	1.5	1	1
2	Ref turun ke 10 cm/s	2.5	2.5	1	1
3	ref naik ke -15cm/s	2	1.5	1	1
4	ref turun ke -10cm/s	2.85	2.5	1	1

Dari data gambar uji kecepatan dengan *acuan* yang diwakili garis merah dan keluaran kecepatan motor yang diwakili garis biru dapat dilihat bahwa dengan menambahkan kendali *close loop* PI pada motor, dapat memperbaiki tanggapan motor. Walaupun tanggapan motor menjadi lebih lambat tetapi kecepatan motor dapat mencapai acuan yang diinginkan. Osilasi yang terjadi berkisar pada jangkauan 1 cm/s - 2 cm/s. Hal ini dikarenakan ketelitian pada pengukuran kurang. Selain itu dalam pengujian motor DC juga dapat dilihat bahwa kecepatan motor DC berosilasi walau dengan nilai masukan yang tetap yang dikarenakan ketidak sempurnaan mekanik dan juga pembacaan *optocoupler* yang kurang teliti.

4.2.2 Pengujian Kendali Sudut dengan Kendali Proporsional

Kendali proporsional digunakan dalam pengaturan sudut hadap robot terhadap sudut hadap acuan. Pengujian kendali proporsional ini adalah dengan memberikan masukan sudut acuan yang tetap pada robot dan melihat tanggapan robot. Pengujian juga akan membandingkan tanggapan robot terhadap sudut acuan bila kecepatan aktual robot dikendalikan secara *close loop* dengan tanggapan robot tanpa kendali kecepatan aktual secara *close loop*. Selain itu dilakukan pengujian terhadap tanggapan robot bila diberikan gangguan sesaat pada robot.

Pengujian sudut hadap robot dilakukan dengan tiga sudut acuan yang berbeda yaitu 45°, 90°, dan 180°.

Tabel 4.4 Hasil uji kendali Proporsional

No.	Masukan (derajat)	Waktu mencapai acuan (s)	Galat Tunak (derajat)	Waktu pulih dari gangguan (s)
1	45	9	2	8.5
2	90	12	2	9.5
3	180	10	1	8.5

4.2.3 Pengujian Posisi Koordinat Akhir Robot

Pada robot, jarak antara sasaran dan posisi robot merupakan parameter yang memutuskan kapan robot bergerak dan kapan robot berhenti. Robot akan bergerak ketika jarak lebih dari 10cm, dan berhenti ketika jarak kurang dari 10cm. Dengan demikian dapat dikatakan jarak merupakan parameter kendali ON-OFF pada robot. Selain pengujian jarak, pada pengujian ini juga diuji algoritma *update* jarak yang digunakan robot. Mekanisme pengujian adalah membandingkan hasil dari posisi akhir koordinat robot (x,y) yang merupakan perhitungan dari program dan posisi akhir koordinat robot pada lintasan uji yang diukur secara manual menggunakan alat ukur (meteran). Posisi koordinat yang dibangkitkan oleh robot didapat dari *encoder* serta perhitungan odometrinya. Hasil perbandingan dapat dilihat pada Tabel 4.5.

Tabel 4.5 Perbandingan hasil posisi akhir robot

Target (cm)		Posisi Akhir Perhitungan (cm)		Posisi Akhir Riil (cm)		Galat akhir robot terhadap jarak Riil (cm)
X	Y	X	Y	X	Y	
120	0	116	2	116	2	4.47
120	120	116	117	116	117	5
0	120	3	116	3	116	5
-120	120	-117	118	-117	118	3.6
-120	0	-118	-5	-118	-4	4.47
-120	-120	-118	-116	-118	-116	4.47
0	-120	-3	-116	-3	-115	5.8
120	-120	116	-116	116	-116	5.6

Dari Tabel 4.5 di atas, didapatkan bahwa robot berhenti tidak tepat pada posisi sasaran. Hal tersebut disebabkan algoritma robot yang dibuat menghentikan robot bila jaraknya kurang dari 10cm dari sasaran. Untuk beberapa pengujian terdapat perbedaan posisi antara hasil perhitungan robot dengan posisi nyata, hal tersebut dikarenakan kemungkinan adanya selip pada robot ketika berotasi. Kekurang telitian *encoder* juga berpengaruh terhadap perhitungan posisi robot.

4.3 Pengujian Tingkah Laku Robot

Pengujian tingkah laku robot dilakukan untuk mengetahui dan menganalisa algoritma sistem pengendalian robot. Pengujian terhadap robot menghasilkan tingkah laku robot sebagai berikut :

- Apabila ketiga sensor jarak tidak mendeteksi adanya halangan, maka gerakan robot bergerak menuju sasaran.
- Robot bergerak lurus bila sasaran berada di depan robot.
- Robot belok ke kanan bila sasaran berada di sebelah kanan robot.
- Robot belok ke kiri bila sasaran berada di sebelah kiri robot.
- Robot berputar ke kanan bila sasaran berada di belakang robot.
- Ketika robot bergerak dan mengetahui adanya halangan di kanan atau kiri robot, maka robot berusaha menghindari dengan berbelok ke arah dimana tidak ada halangan.
- Ketika robot bergerak dan mengetahui adanya halangan di depan robot, maka robot mengurangi kecepatan berusaha menghindari ke arah dimana tidak ada halangan. Apabila arah kanan dan kiri tidak terdapat halangan, maka robot menghindari ke kiri.
- Ketika robot selesai menghindari rintangan robot tidak langsung berbelok mendadak ke sasaran, tetapi berbelok secara perlahan menuju sasaran.
- Robot berhenti bila jarak terhitung kurang dari 10 cm dari sasaran.
- Robot berhenti bila ada rintangan di depan, kanan dan kiri sekaligus pada jarak 20cm dan menampilkan tulisan “ *i'am stack here! mision failed!!!*” pada LCD.

V PENUTUP

5.1 KESIMPULAN

Berdasarkan perancangan, pengujian dan analisa yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Penggunaan dua motor yang tidak identik dan kurang linier dapat mengganggu algoritma pengendalian robot tipe diferensial.
2. Pengendalian kecepatan aktual roda dengan metode proporsional integral dapat memperbaiki kelemahan motor penggerak robot yang tidak identik satu sama lain.

3. Pengendalian kecepatan secara *close loop* dapat memperlambat tanggapan motor terhadap masukannya.
4. Sistem kendali Proporsional mampu mengendalikan pergerakan robot mobil untuk mencapai tempat sasaran.
5. Odometri mampu memberikan perhitungan posisi yang baik untuk jarak yang pendek.
6. Pengaturan sensor PING™ berpengaruh terhadap tanggapan robot dalam menghindari rintangan.
7. Penggunaan filter orde satu pada kecepatan angular robot dapat memperhalus pergerakan robot.

5.1 Saran

Untuk pengembangan sistem lebih lanjut, maka penulis memberikan saran-saran sebagai berikut:

1. Kita dapat menambahkan lagi sensor PING)))™ untuk memperbaiki jangkauan sensor dalam mendeteksi rintangan
2. Sensor posisi yang digunakan lebih baik menggunakan sensor global, sebagai contoh dengan *beacon* ultrasonik, GPS, atau sensor kompas untuk mengeliminasi kesalahan posisi di lapangan.

DAFTAR PUSTAKA

- [1] Heryanto, M.Ary & Wisnu Adi P., *Pemrograman Bahasa C untuk Mikrokontroler ATMEGA8535*, Penerbit Andi, Yogyakarta, 2008.
- [2] Kadir, Abdul, *Pemrograman C++*, Penerbit Andi, Yogyakarta, 1995.
- [3] Setiawan, Iwan, Trias A., Darjat, *Rancang Bangun Sistem Kontrol Robot Mobil untuk Keperluan Navigasi Darat Berbasis Trayektori Bezier*, Universitas Diponegoro, 2006.
- [4] Orgen, Petter, *Doctoral Thesis : Formation and Obstacle Avoidance in Mobile Robot Control*, Stockhlom, 2003.
- [5] Jacquot, Raymond G., *Modern Digital Control Systems*, Marcel Dekker, Inc. New York and Basel, 1981.
- [6] Ogata, Katsuhiko, *Teknik Kontrol Automatik Jilid 1-2*, Diterjemahkan Oleh Ir. Edi Leksono, Penerbit Erlangga, Jakarta, 1996.
- [7] Bogoni, Luca, “*Subsumption Architecture and Discrete Event Systems: A Comparison*”, Department of Computer and Information

Science University of Pennsylvania, Philadelphia, 1993.

- [8] Brooks, R. A., “*A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation*”, 2, 14–23, 1986.
- [9] Harper, Christopher J. dan Alan F. T. Winfield, “*Designing Intelligent Control Systems for Safety Critical Applications*”, University of the West of England, Inggris.
- [10] Craig, John J., *Introduction to Robotics Mechanism and Control*, Addison-Wesley, 1986.
- [11] -----, *Atmega32 Data Sheet*, <http://www.atmel.com>.
- [12] -----, PING)))™ Ultrasonic Distance Sensor *Data Sheet*, <http://www.parallax.com>.
- [13] -----, L298 *Data Sheet*, <http://www.ti.com>.

R. Kurniwan Indra L. (L2F 004 501)



Lahir di Semarang, 1 Maret 1986. Saat ini sedang melanjutkan studi pendidikan strata I di Jurusan Teknik Elektro, Fakultas Teknik Universitas Diponegoro Konsentrasi Kontrol.

Mengetahui dan mengesahkan,

Dosen Pembimbing I

Dosen Pembimbing II

Iwan Setiawan, ST, MT

NIP. 132 283 183

Tanggal: _____

Darjat, ST, MT

NIP. 132 231 135

Tanggal: _____