

PENGENDALIAN ROBOT MOBIL PENCARI TARGET DAN PENGHINDAR RINTANGAN DENGAN METODE KONTROL PD MENGGUNAKAN MIKROKONTROLER AVR ATMEGA 8535

Anis Zaenal Abidin¹, Iwan Setiawan, ST, MT², Trias Andromeda, ST, MT²
Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang
e-mail : An_nistelrooy@yahoo.com

Abstrak

Robotika merupakan bidang ilmu pengetahuan yang mengalami perkembangan dengan cepat, salah satunya adalah Wheeled Mobile Robot (WMR). Pembahasan mengenai robot mobil meliputi banyak hal seperti fungsi dan kegunaannya, sistem navigasinya, metode kontrol yang digunakan dan lain-lain. Sedangkan contoh aplikasinya dalam kehidupan sehari-hari meliputi robot pembersih ruangan, penjinak bom, kursi roda cerdas (Smart Wheel Chair), Self Parking Car (SPC), hingga penghantar barang di pabrik-pabrik ataupun rumah sakit. Untuk sistem navigasi robot mobil, banyak peneliti yang telah mengembangkan robot mobil beroda yang dapat melakukan navigasi tanpa garis sebagai guidance dengan menghitung posisi dan orientasinya berdasarkan informasi putaran rodanya yang biasa dikenal dengan istilah *dead reckoning*.

Tugas Akhir ini bertujuan mengimplementasikan kendali PD pada robot mobil pencari target dan penghindar rintangan. Kendali PD digunakan untuk mengendalikan kecepatan roda kanan dan roda kiri robot mobil. Sedangkan pencarian target digunakan metode odometri dengan bantuan encoder untuk menentukan posisi dan orientasi dari robot mobil. Dan digunakan sensor ultrasonik sebagai pendeteksi ada tidaknya rintangan. Dengan demikian robot mobil mampu berjalan menuju target tanpa menabrak rintangan yang menghalanginya.

Dari hasil penelitian didapatkan bahwa ketelitian pembacaan encoder sangat berpengaruh pada posisi dan orientasi robot mobil. Semakin banyak halangan yang terdeteksi maka error posisi akhirnya semakin besar. Parameter pengendali juga berpengaruh pada gerakan robot, jika nilai K_p terlalu besar maka robot mudah berosilasi. Dari hasil pengujian diketahui bahwa posisi sensor ultrasonik sangat berpengaruh terhadap pendeteksian halangan.

Kata kunci : Robot mobil, Odometri, Pengendali PD, Sensor Ultrasonik

I. PENDAHULUAN

1.1 Latar Belakang

Banyak peneliti yang telah mengembangkan *wheeled mobile robot* (wmr) atau robot mobil beroda yang dapat melakukan navigasi tanpa garis sebagai *guidance*. Beberapa *wheeled mobile robot* melakukan navigasi dengan menghitung posisi dan orientasinya berdasarkan informasi putaran rodanya yang biasa dikenal dengan istilah *dead reckoning*. Peneliti lain telah mengembangkan teknik-teknik agar robot dapat menghindari halangan atau biasa dikenal dengan *obstacle avoidance*.

Pada Tugas Akhir ini, selain mempunyai kemampuan menghindari rintangan robot mobil juga mempunyai kemampuan untuk menuju target posisi yang diinginkan. Pengendali PD (*Proportional-Derivative*) digunakan untuk mengontrol pergerakan robot. Pada dasarnya pengendali proporsional sudah mencukupi untuk mengendalikan pergerakan robot mobil. Penambahan pengendali *derivative* ini disebabkan karena sifat dari pengendali *derivative* yang dapat membantu memperbaiki *error* dan memprediksi *error* yang akan terjadi.

1.2 Tujuan

Tujuan dari penelitian Tugas Akhir ini adalah membangun suatu sistem navigasi robot mobil

dengan kemampuan menghindari rintangan dalam upaya menuju target posisi yang diinginkan.

1.3 Pembatasan Masalah

Batasan masalah pada Tugas Akhir ini adalah sebagai berikut :

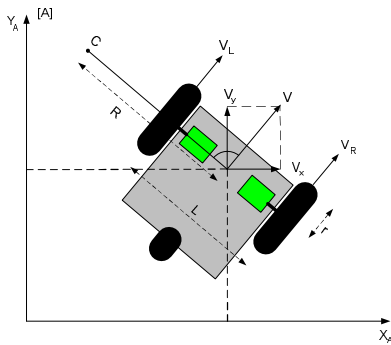
1. Robot mobil yang dibuat menggunakan tipe penggerak diferensial.
2. Pergerakan robot mengabaikan massa, percepatan, gaya, gesekan karena robot mobil yang dibuat berukuran kecil.
3. Mikrokontroler yang digunakan adalah mikrokontroler AVR ATmega8535.
4. Sensor pendeteksi halangan sekaligus sensor jarak yang digunakan adalah sensor ultrasonik (PINGTM).
5. Pendeteksian sensor ultrasonik dibatasi pada jarak 3 – 255 cm.
6. Digunakan 2 buah sensor ultrasonik, sehingga terdapat keterbatasan dalam pendeteksian halangan.
7. Robot melakukan aksi menghindari rintangan dengan cara berbelok arah ke kanan atau ke kiri.
8. Robot mobil hanya dapat dioperasikan bila tidak terdapat sumber gelombang ultrasonik lain.
9. Pengendali yang digunakan adalah pengendali proposional plus diferensial (PD) dengan tuning parameter dilakukan secara empiris.
10. Pengendalian kecepatan motor DC dilakukan melalui PWM.

1) Mahasiswa Jurusan Teknik Elektro UNDIP
2) Staf Pengajar Jurusan Teknik Elektro

II. DASAR TEORI

2.1 Kinematika Robot Mobil Penggerak Differensial

Salah satu jenis robot mobil yang umum digunakan, terutama untuk dioperasikan dalam ruang adalah robot mobil dengan sistem penggerak diferensial (*differential drive*). Alasannya karena relatif lebih fleksibel dalam melakukan manuver serta kemudahan dalam pengontrolannya. Gambar 1 memperlihatkan arsitektur robot tersebut.



Gambar 1 Posisi dan Orientasi robot mobil dalam sistem koordinat cartesian

Untuk panjang jari-jari roda r , serta kecepatan rotasi roda kanan dan kiri ω_R dan ω_L , maka kecepatan linier roda kanan dan kiri dapat dicari dengan persamaan berikut.

$$V_R(t) = r\omega_R(t) \dots\dots\dots (2.1)$$

$$V_L(t) = r\omega_L(t) \dots\dots\dots (2.2)$$

Kecapatan rotasi robot ketika melakukan gerak memutar sesaat dengan jari-jari R adalah

$$\omega_R = V_R / (R + L/2) \dots\dots\dots (2.3)$$

$$\omega_L = V_L / (R - L/2) \dots\dots\dots (2.4)$$

Sedangkan kecepatan rotasi robot keseluruhan adalah,

$$\omega(t) = \frac{(V_R(t) - V_L(t))}{L} \dots\dots\dots (2.5)$$

Dan jari-jari lintasan robot dan kecepatan liniernya dapat dicari dengan Persamaan berikut,

$$R = \frac{L(V_R + V_L)}{2(V_R - V_L)} \dots\dots\dots (2.6)$$

$$V(t) = (V_R(t) + V_L(t)) / 2 \dots\dots\dots (2.7)$$

Bentuk matrik-vektor Persamaan (2.5) dan (2.7) adalah,

$$\begin{bmatrix} V(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} V_R(t) \\ V_L(t) \end{bmatrix} \dots\dots\dots (2.8)$$

$$\begin{bmatrix} V_R(t) \\ V_L(t) \end{bmatrix} = \begin{bmatrix} 1 & \frac{L}{2} \\ 1 & -\frac{L}{2} \end{bmatrix} \begin{bmatrix} V(t) \\ \omega(t) \end{bmatrix} \dots\dots\dots (2.9)$$

Dengan mengetahui kecepatan linier dan angular robot ini setiap saat, maka kecepatan pada setiap sumbu kartesian dapat dicari dengan cara memproyeksikan vektor kecepatan robot kepada sumbu-sumbu tersebut.

2.2 Pengendali PID

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

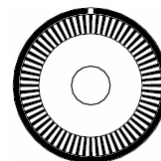
$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \dots\dots\dots (2.10)$$

2.3 Mikrokontroler ATmega8535

ATmega8535 merupakan mikrokontroler AVR berbasis arsitektur RISC, yang sebagian besar instruksinya dieksekusi dalam 1 (satu) siklus *clock*. AVR ATmega8535 mempunyai 2 jenis *Timer* yang cara kerjanya berlainan, yakni *Timer* 8 bit dan *Timer* 16 bit. Frekuensi untuk kedua macam *Timer* ini bisa diatur terpisah. Sebagai *Timer*, frekuensi penggerak berasal dari osilator kristal yang frekuensinya diturunkan dengan faktor 1X, 8 X, 64X, 256X, atau 1024X. Selain dipakai sebagai *Timer/Counter*, *Timer* AVR bisa dipakai untuk keperluan pembangkitan sinyal PWM (*Pulse Width Modulation*) secara perangkat keras, artinya cukup mengatur beberapa register untuk keperluan PWM. Sinyal PWM akan disalurkan ke kaki OC0, OC1A, OC1B, atau OC2. Pada dasarnya sinyal PWM merupakan semacam *digital to analog converter* pada kaki OC0, OC1A, OC1B, atau OC2, di mana nilai tegangan analognya bisa ditentukan lewat program.

2.4 Odometri

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



Gambar 2 Ilustrasi piringan encoder

Odometri memerlukan sebuah metode untuk menghitung dengan akurat perputaran dari roda robot. Metode standar untuk melakukan ini adalah dengan memasang roda dengan *encoder* optik

seperti pada Gambar 2. 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.

$$RC = \frac{\text{Jumlah Lubang Encoder}}{\text{Keliling Lingkaran}} \dots\dots\dots(2.11)$$

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

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

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

$$\text{Jarak} = (\text{Jarak}_R + \text{Jarak}_L) / 2 \dots\dots\dots(2.12)$$

$$\theta = \theta_0 + \frac{(\text{Jarak}_R - \text{Jarak}_L)}{L} \dots\dots\dots(2.13)$$

$$X = X_0 + \text{Jarak} * \cos(\theta) \dots\dots\dots(2.14)$$

$$Y = Y_0 + \text{Jarak} * \sin(\theta) \dots\dots\dots(2.15)$$

Keterangan :

Jarak_R = Jarak tempuh roda kanan robot

Jarak_L = Jarak tempuh roda kiri robot

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

L = Jarak antara roda kanan dan kiri

Jarak = Jarak yang ditempuh oleh robot

θ = Sudut atau orientasi yang dibentuk oleh robot dalam radian

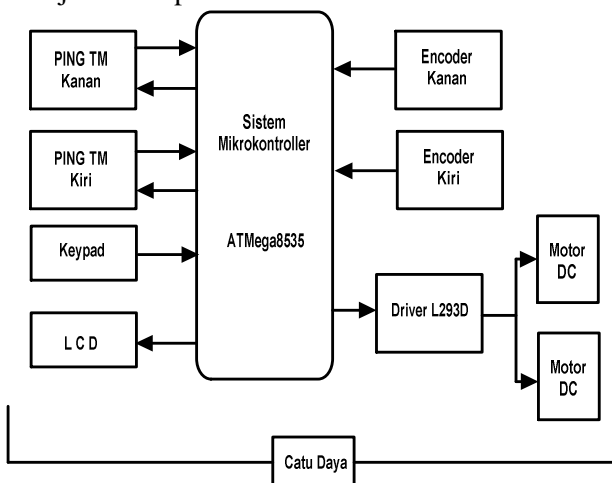
X dan Y = Posisi robot dalam absis dan ordinat

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

III. PERANCANGAN

3.1 Perancangan Perangkat Keras

Diagram blok perancangan perangkat keras, ditunjukkan pada Gambar 3 berikut.



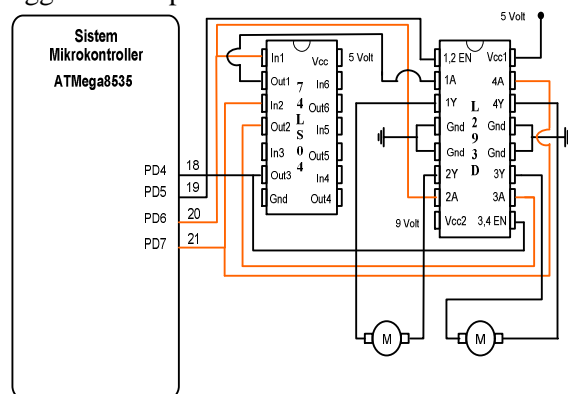
Gambar 3 Blok diagram sistem

Secara umum perancangan sistem ini dapat dijelaskan sebagai berikut :

1. Mikrokontroler ATmega8535 sebagai pusat pengelolaan data yang diprogram menggunakan bahasa C embedded.
2. Dua buah sensor Ping sebagai pendeteksi rintangan sekaligus mengukur jarak rintangan tersebut.
3. Dua buah motor dc sebagai penggerak roda kanan dan roda kiri robot. Kecepatan putarnya dikendalikan melalui perubahan *duty cycle* PWM.
4. *Driver* motor dc L293D untuk memnguatkan arus keluaran dari mikrokontroler.
5. Sensor *Optocoupler* digunakan untuk mendeteksi jumlah lubang pada piringan *encoder*.
6. keypad digunakan untuk memberi masukan yang berupa posisi absis dan ordinat (x,y) target serta parameter kontrol (K_p dan K_d).
7. LCD digunakan sebagai komunikasi antara program dan pemakai.

3.1.1 Driver Motor L293D

Driver motor dc menggunakan IC L293D yang mampu untuk menjalankan motor dengan arus hingga 600mA per *channel*.



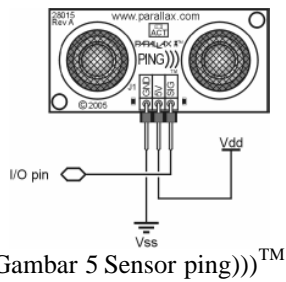
Gambar 4 Rangkaian *Driver* motor L293D

Keluaran sinyal PWM dari mikrokontroler ada pada PIND.4 dan PIND.5, sinyal PWM akan mengatur tegangan masukan ke motor dengan mengatur lebar pulsanya. Sedangkan PIND.6 dan PIND.7 digunakan sebagai kombinasi untuk menentukan arah perputaran motor. Untuk satu motor digunakan dua masukan arah yang berbeda yaitu (logika 0 dan 1) oleh karena itu untuk penghematan pemakaian pin dari mikrokontroler digunakan IC 74LS04 sebagai *inverter*.

3.1.4 Sensor Ultrasonik

Kebutuhan dasar semua robot mobil otomatis adalah kemampuan menghindari rintangan yang dapat membantu robot mobil bergerak tanpa benturan di dalam lingkungan yang diinginkan.

Sensor ultrasonik dapat digunakan untuk mendeteksi ada tidaknya halangan.



Gambar 5 Sensor ping)))TM

Untuk memulai pengukuran sensor perlu dipicu dengan sinyal *high* selama t_{OUT} , kemudian menunggu selama $t_{HOLDOFF}$ baru sensor memancarkan sinyal ultrasonik dan siap untuk menerima kembali sinyal pantulan. Waktu yang diperlukan gelombang ultrasonik mulai dari dipancarkan sampai diterima kembali menentukan besarnya jarak.

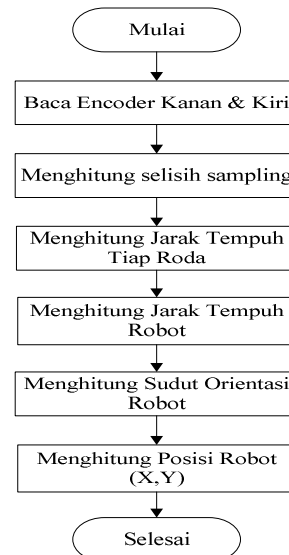
3.2 Perancangan Perangkat Lunak

Robot yang dibuat pada Tugas Akhir ini adalah robot yang mempunyai tugas untuk mencapai target yang diinginkan dan juga berkemampuan untuk menghindari rintangan yang ada selama robot menuju target tersebut. Target berupa koordinat kartesian (x,y), dengan sumbu x positif mewakili arah maju robot, dan sumbu y positif mewakili arah kiri robot. Program utama dari perancangan robot mobil ini meliputi tiga subrutin program utama, yaitu :

1. Odometri
Merupakan subrutin program untuk meng-update posisi dan orientasi robot saat ini.
2. *Go to Goal*
Merupakan subrutin program yang berisi perintah untuk mengatur robot supaya tetap pada jalurnya menuju target.
3. *Avoiding Static Obstacle*
Merupakan subrutin program yang berisi perintah untuk mengatur robot agar tidak berbenturan dengan halangan yang berada di depannya.

3.2.1 Odometri

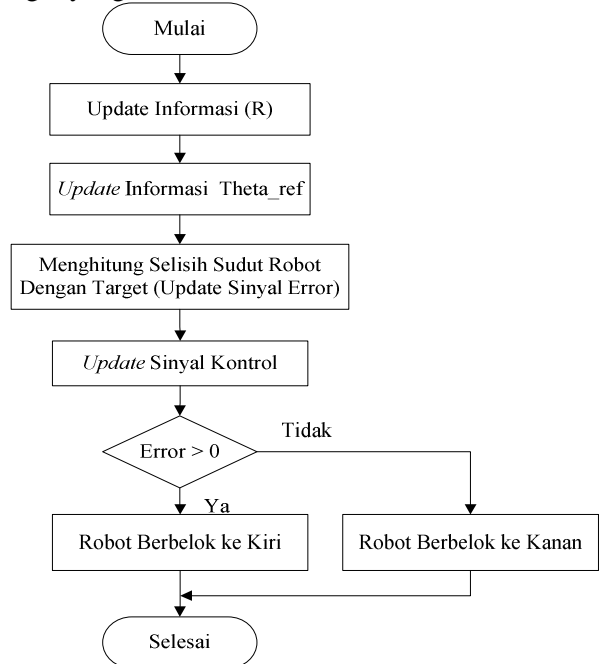
Sensor yang digunakan adalah sensor *optocoupler* untuk masing-masing roda untuk mendeteksi banyaknya lubang pada piringan *encoder* Kemudian banyaknya lubang per cm yang terdeteksi diolah untuk menghasilkan informasi posisi dan orientasi dari robot saat ini, dan rutin ini dikerjakan secara berulang-ulang atau selalu diperbaharui dengan periode tertentu. *Flowchart* dari subrutin odometri dapat dilihat pada Gambar 12 berikut.



Gambar 6 Diagram Alir Rutin Odometri

3.2.2 *Go to Goal*

Tugas subrutin ini adalah memerintahkan robot untuk tetap berada pada jalurnya menuju target yang telah ditentukan.



Gambar 7 Flowchart subrutin *Go to Goal*

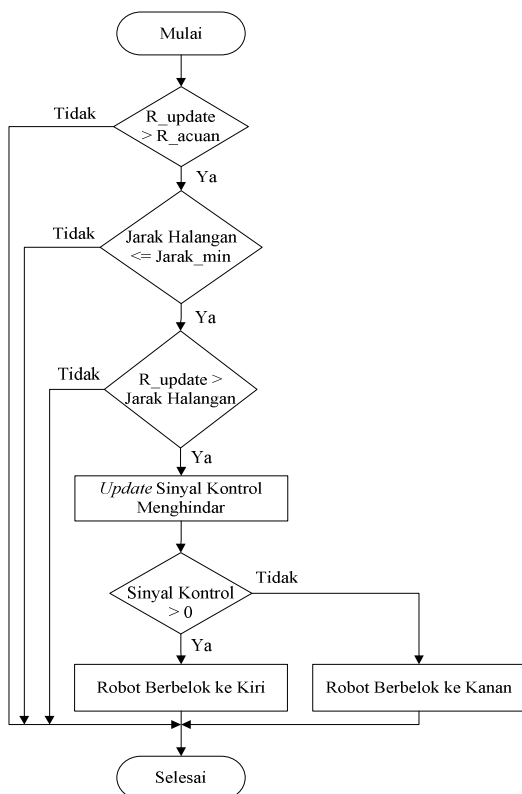
Flowchart dari subrutin ini dapat dilihat pada Gambar 7 di atas. Proses *Go to Goal* ini menggunakan informasi *error* sudut orientasi sebagai pengendali agar robot tetap menuju ke target. Target didapatkan dari koordinat X dan Y yang dimasukkan oleh *user*, dari informasi ini kemudian didapatkan sudut referensi robot, yaitu sudut yang terbentuk antara arah hadap robot dengan tujuan (X,Y), dan juga didapatkan informasi jarak target dengan robot sekarang (R).

3.2.3 Avoiding Static Obstacle

Obstacle Avoiding merupakan aksi menghindari yang dilakukan oleh robot apabila terdapat rintangan pada jarak tertentu yang menghalangi robot menuju target. Aksi ini dilakukan jika syarat-syarat posisi halangan berikut ini dipenuhi :

1. Jika jarak halangan yang terdeteksi oleh sensor PING lebih kecil atau sama dengan jarak minimal di mana robot harus mulai menghindari (ditentukan *programmer*). Atau dengan kata lain halangan berada pada jarak kurang dari jarak minimal yang ditentukan oleh *programmer* (50cm).
2. Jika jarak robot terhadap target lebih besar dari jarak halangan yang terdeteksi.

Jika kedua syarat di atas terpenuhi, maka robot akan menghasilkan sinyal kontrol untuk memulai aksi menghindari.



Gambar 8 Flowchart subrutin *Avoiding Static Obstacle*

Arah hindaran robot tergantung jarak rintangan yang terdeteksi, jika rintangan berada di sebelah kanan robot menghindari ke kiri dan sebaliknya. Diagram alir dari proses subrutin ini adalah sebagai berikut. Untuk menghasilkan sinyal kontrol supaya robot menghindari digunakan persamaan di bawah ini;

$$U(k) = Gain * \left(\frac{1}{ping_R} - \frac{1}{ping_L} \right) * \frac{1}{(ping_R - ping_L)}$$

keterangan :

Gain = konstanta penguat sinyal kontrol

ping_R = jarak halangan yang terdeteksi sensor PING kanan

ping_L = jarak halangan yang terdeteksi sensor PING kiri

Persamaan di atas didapat dari proses *trial and error* untuk mendapatkan respon berupa sinyal kontrol untuk aksi menghindari yang cocok, yaitu semakin dekat halangan maka dibutuhkan sinyal kontrol yang semakin besar.

IV. ANALISA DAN PENGUJIAN

4.1 Pengujian Perangkat Lunak

Pengujian perangkat lunak yang dilakukan meliputi pengujian kendali PD (Proporsional – Differensial) pada pengontrolan sudut orientasi untuk mendapatkan parameter kontrol (*Kp* dan *Kd*) yang optimal. Dan juga dilakukan pengujian pada posisi akhir robot, dengan cara membandingkan koordinat akhir yang dibangkitkan oleh robot yang ditampilkan di LCD dengan koordinat akhir robot pada lintasan uji yang diukur secara manual menggunakan alat ukur (meteran). Pengujian posisi akhir robot yang dibangkitkan oleh robot dilakukan untuk robot tanpa halangan dan robot dengan halangan.

4.2 Pengujian Perangkat Keras

4.2.1 Pengujian Driver Motor DC

Driver motor L293D ini mendapat masukan dari PWM yang dihasilkan oleh *timer* 1 mikrokontroler AVR. Mode PWM internal yang digunakan adalah *Phase and Frequency Correct* PWM *top=0xFF*. Keluaran sinyal PWM didapat pada PIND.4 dan PIND.5 dengan cara mengatur nilai dari register OCR1AL dan OCR1BL.

Hasil pengujian *driver* motor dc L293D dapat dilihat pada Tabel 4.1 berikut. Hasilnya terlihat bahwa semakin besar nilai OCR1AL dan OCR1BL yang diberikan maka akan semakin besar tegangan yang dihasilkan.

Tabel 4.1 Hasil Pengujian *Driver* L293D

Nilai OCR1AL	Nilai OCR1BL	V _{keluaran channel 1 (V) terukur}	V _{keluaran channel 2 (V) terukur}	V _{keluaran (V) perhitungan}
255	255	8,9	8,9	9
225	225	7,8	7,8	7,9
200	200	6,9	6,9	7,05
175	175	6,1	6,1	6,2
150	150	5,2	5,2	5,3
125	125	4,2	4,2	4,4
100	100	3,4	3,4	3,5
75	75	2,5	2,5	2,6

4.2.2 Pengujian Sensor Ping

Pengujian terhadap sensor PING dilakukan dengan mengukur jarak suatu benda terukur yang diletakkan di hadapan sensor, pembacaan sensor dibatasi pada jarak 3 cm – 255 cm. Data hasil pengukuran yang dilakukan adalah sebagai berikut.

Tabel 4.2 Hasil pengukuran jarak dengan PING™.

No.	Jarak Terukur (cm)	Jarak Terdeteksi PING™ (cm)
1	10	9
2	20	19
3	30	28
4	40	38
5	50	48
6	60	58
7	70	66
8	80	76

4.3 Perbandingan Posisi Koordinat Akhir Robot

Dalam pengujian ini akan dibandingkan hasil dari posisi akhir koordinat robot (x,y) yang merupakan perhitungan dari program dan ditampilkan di LCD 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. Pengujian ini dibedakan menjadi dua yaitu perbandingan posisi koordinat akhir robot tanpa halangan dan perbandingan posisi koordinat akhir dengan halangan. Sedangkan konstanta pengontrolan yang digunakan dalam pengujian ini adalah konstanta optimal yang didapat dari pengujian sebelumnya, yaitu $K_p = 10$ dan $K_d = 5$.

Dari Tabel 4.3 berikut, didapatkan bahwa robot dapat mencapai target yang diinginkan. Kesalahan posisi disebabkan karena kurang telitian dalam pembacaan jumlah pulsa *encoder*. Semakin banyak jumlah halangan semakin besar *error* yang dihasilkan, ini karena semakin banyak interupsi yang terjadi dalam program.

Tabel 4.3 Perbandingan hasil posisi akhir robot

Target (cm)		Jumlah halangan	Posisi Akhir Riil (cm)		Posisi Akhir Perhitungan (cm)	
X	Y		X	Y	X	Y
300	0	0	294	-1	298	2
		1	292	4	296	-1
		2	290	-3	295	1
		3	290	-2	296	3
100	100	0	96	97	96	98
		1	94	95	96	95
		2	92	90	96	96

		3	94	90	95	96
100	-100	0	95	-96	96	-95
		1	95	-95	97	-98
		2	96	-94	95	-96
		3	93	-94	96	-97
0	100	0	1	97	-1	98
		1	2	96	0	97
		2	3	96	2	97
		3	3	95	2	96
0	-100	0	1	-96	0	-97
		1	3	-96	2	-96
		2	3	-95	2	-98
		3	4	-95	3	-96
-200	-200	0	-195	-195	-195	-198
		1	-193	-192	-196	-196
		2	-192	-195	-195	-195
		3	-190	-194	-196	-196

V. PENUTUP

5.1 Kesimpulan

Dari perancangan, pembuatan dan pengujian pada Tugas Akhir mobil robot pencari target dan penghindar rintangan dengan menggunakan kendali Proporsional – Differensial ini bisa ditarik kesimpulan sebagai berikut :

1. Dari hasil pengujian kendali kecepatan motor tanpa beban, didapatkan bahwa untuk sistem tanpa kontroler menghasilkan respon dengan *error* yang besar.
2. Pengujian dengan kontroler proporsional (K_p) mempercepat keluaran respon menuju nilai referensi. Semakin besar nilai K_p , maka semakin cepat *rise time* nya. Tetapi untuk K_p yang terlalu besar (dari hasil percobaan, $K_p > 15$) akan menyebabkan sistem berosilasi.
3. Kontroler differensial (K_d) berpengaruh pada sensitivitas sistem terhadap perubahan sudut yang dibentuk oleh robot.
4. Ketelitian dari sensor *optocoupler* sangat berpengaruh terhadap pengaturan posisi dan orientasi dari robot.
5. Posisi dan arah sensor ultrasonik sangat berpengaruh terhadap pendeteksian rintangan yang berada di depan robot.
6. Semakin banyak halangan yang diterima robot dalam menuju target, maka *error* posisinya akan semakin besar. Hal ini berkaitan dengan sensitivitas sensor.

5.2 Saran

Sebagai masukan guna pengembangan lebih lanjut dari Tugas Akhir ini, maka penulis memberikan beberapa saran sebagai berikut :

1. Untuk mendapatkan nilai parameter-parameter K_p dan K_d yang optimal dapat digunakan metode tuning parameter.
2. Untuk mendapatkan respon yang lebih bagus dari PD dapat digunakan pengendali yang lain seperti sistem kontrol adaptif atau sistem kontrol cerdas.
3. Pembacaan sensor *optocoupler* dan ultrasonik dapat digunakan unit pemroses tersendiri, sehingga perhitungan program utama dan *update* PWM tidak terganggu.
4. Penambahan sensor ultrasonik akan menjadikan robot lebih sensitif dalam membaca adanya halangan.
5. Untuk mendapatkan posisi yang lebih akurat dapat digunakan sensor kompas magnetik.



Anis Zaenal Abidin

Lahir di Semarang, menyelesaikan pendidikan di SMU Islam Hidayatullah Semarang, dan melanjutkan pendidikan S1 di UNDIP Semarang. Saat ini sedang melanjutkan studinya di jurusan Teknik Elektro dengan konsentrasi Kontrol.

Menyetujui / Mengesahkan,

Pembimbing I,

Pembimbing II,

Iwan Setiawan, ST, MT.
NIP. 132 283 183

Trias Andromeda
NIP. 132 283 185

DAFTAR PUSTAKA

- [1] Anderson, David, *IMU Odometry*, <http://www.geology.smu.edu/%7Edpa-www/dpa.html>, Texas, 2006
- [2] Borenstein, J; Everett, H. R. and Feng, L, Contributing authors: S. W. Lee and R. H. Byrne, *Where am I ? Sensors and Methods for Mobile Robot Positioning*, Michigan, 1996.
- [3] Budiharto, Widodo, *Membuat Robot Cerdas*, Penerbit Elex Media Komputindo, Jakarta, 2006.
- [4] Halawa, Edward E.H dan Sakti, Setyawan P, *Pemrograman Dengan C/C++ dan Aplikasi Numerik*, Penerbit Erlangga, Jakarta, 1995.
- [5] Ogata, Katsuhiko, *Teknik Kontrol Automatik Jilid 1-2*, Diterjemahkan Oleh Ir. Edi Leksono, Penerbit Erlangga, Jakarta, 1996.
- [6] Setiawan, Iwan, *Rancang Bangun Sistem Kontrol Robot Mobil untuk Keperluan Navigasi Darat Berbasis Trayektori Bezier*, Universitas Diponegoro, 2006.
- [7] Wardhana L, *Belajar Sendiri Mikrokontroler AVR Seri ATmega8535 Simulasi, Hardware, dan Aplikasi*, Penerbit Andi, Yogyakarta, 2006.
- [8] -----, PING)))™ Ultrasonic Distance Sensor *Data Sheet*, <http://www.parallax.com>.
- [9] -----, ATmega8535 *Data Sheet*, <http://www.atmel.com>.
- [10] -----, L293D *Data Sheet*, <http://www.ti.com>.