

## PERANCANGAN DAN IMPLEMENTASI SISTEM *OBSTACLE AVOIDANCE* PADA *MOBILE ROBOT* BERODA BERBASIS KENDALI PROPORSIONAL

### (SCHEME AND IMPLEMENTATION OF *OBSTACLE AVOIDANCE* SYSTEM IN *WHEELED MOBILE ROBOT* PROPORTIONAL CONTROL BASED)

Abdul Haffif<sup>1</sup>Drs. Suwandi, M.si<sup>2</sup>Ahmad Qurthobi, ST., MT.<sup>3</sup><sup>1-3</sup> Prodi S1 Teknik Fisika, Fakultas Teknik, Universitas Telkom<sup>1</sup>abdulhaffif@gmail.com<sup>2</sup>suwandi.sains@gmail.com<sup>3</sup>qurthobi@gmail.com

#### 1. ABSTRAK

Pada penelitian ini, dibuat suatu implementasi kontrol *Proporsional* pada *mobile robot* untuk menjalankan skema *obstacle avoidance*. *Mobile robot* menggunakan Sensor HC SR04 dan CMPS 10, XBee seri 1 untuk komunikasi data, Arduino sebagai modul mikrokontroler dan *driver motor* sebagai pengubah PWM menjadi arus untuk motor DC. Data yang menjadi kunci penting dalam percobaan ini adalah pembacaan jarak, dan sudut oleh sensor pada robot. Kedua data tersebut akan dikirim ke *station ground* lalu diolah dan outputnya berupa data PWM untuk robot. Sistem kestabilan robot akan ditingkatkan dengan kontrol Proporsional. Nilai fungsi transfer hasil pemodelan adalah  $\frac{\omega_m(s)}{E(s)} = \frac{716000}{s+60400}$ , orde satu. Nilai kontrol yang paling baik adalah  $K_p = 1.5048$ , dimana respon transienya seperti nilai *rise time*, *settling time*, *overshoot*, dan *error steady state* menjadi lebih kecil yang menandakan sistem lebih cepat. Pada saat menghindari tabrakan terjadi penyimpangan nilai jarak dari set point 50 cm, 70 cm, dan 80 cm yaitu sebesar masing-masing 1.27 cm, 2.32 cm, 1.79cm, ini disebabkan oleh adanya delay waktu pengiriman dan pemrosesan data pada transmitter juga receiver, namun robot tetap dapat bergerak menghindari tabrakan dan kembali bergerak lurus dengan stabil menghadap ke arah yang sama seperti awal.

**Kata kunci :** *Proporsional*, Arduino, HC SR04, CMPS 10, PWM, *.Xbee Series1*.

#### ABSTRACT

*In this research the PID control is implemented in mobile robot to run the scheme of obstacle avoidance. The mobile robot uses HC SR04 and CMPS 10 as the sensor, XBee seri 1 as data communicator, Arduino as the microcontroller module and the motor driver as the changer from PWM to electricity current for the DC motor. the important key of this experiment are the data of distance and angle. The data will be transmitted to the station ground, then being treated and the output is the PWM data for the robot. The robot stability system will be increased by the Proportional controller. It is found that the transfer function is*

*$\frac{\omega_m(s)}{E(s)} = \frac{716000}{s+60400}$ , first order. it is obtained that the best control point is at  $K_p=1.5048$ , where the transient response like rise time, settling time, overshoot, and error steady state become lesser. When robot avoiding the obstacle the distance error occur from the set point of 50 cm, 70 cm, dan 80 cm, each of them is 1.27 cm, 2.32 cm, 1.79cm, it is caused by the delay time of transmitting and processing the data in transmitter also in receiver, but somehow robot can get rid from the collision with the obstacle and go straight stably.*

**Keywords :** *Proportional*, Arduino, HC SR04, CMPS 10, PWM, *.Xbee Series1*.

#### 1. PENDAHULUAN

*Mobile robot* adalah robot dengan kemampuan bergerak di sekitar lingkungan dan tidak menetap pada satu lokasi fisik [1]. Menghindari tabrakan (*obstacle avoidance*) adalah salah satu isu utama pada *mobile robot*, dalam aplikasi untuk perindustrian, pendukung aktivitas manusia, dan lain-lain [2]. Pada pengaplikasiannya *mobile robot* baik digunakan di perindustrian, agar robot dapat memiliki lebih banyak fungsi dan efisien. Selain itu sistem *obstacle avoidance* juga berguna sebagai *prototipe* sistem keamanan pada kendaraan.

Untuk mewujudkan suatu sistem yang dapat bekerja secara otomatis maka diperlukan suatu pengontrolan dimana dalam penelitian ini digunakan kontrol proporsional yang merupakan bagian dari PID karena responnya yang cepat dan mampu kolaborasi juga dengan sistem kontrol lain. Sehingga bisa dikembangkan lebih lanjut.

Adapun tujuan dari penelitian ini adalah membuat suatu sistem *mobile robot* yang dapat menghindari robot yang bergerak dari arah berlawanan pada satu lajur, dan robot dapat bergerak dengan stabil dan menurunkan kecepatan untuk selanjutnya berbelok pada jarak tertentu (terdekat). Robot diatur untuk berbelok ke kiri dan membentuk radius sekecil mungkin antar robot. Setelah menghindar berhasil dilakukan robot dapat kembali bergerak lurus dengan stabil.

Beberapa rumusan masalah yang menjadi acuan dalam penelitian ini adalah, bagaimana fungsi transfer motor DC hasil pemodelan, bagaimana respon transien dari fungsi transfer motor DC sebelum dan sesudah pengontrolan, dan Apakah robot menghindari pada *set point* jarak yang telah ditentukan yaitu 50 cm, 70 cm, dan 80 cm

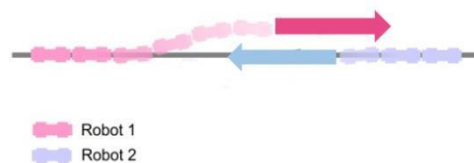
## 2. TEORI DASAR DAN PERANCANGAN SISTEM

### 2.1 Mobile Robot

*Mobile* robot adalah robot yang ciri konstruksinya menggunakan roda atau kaki sebagai aktuatornya untuk menggerakkan keseluruhan badan robot, sehingga robot tersebut dapat berpindah dari satu tempat ke tempat lain dengan sistem navigasi yang berasal dari sensor. Roda sebagai daya penggerak sudah sangat populer digunakan oleh banyak orang karena mudah dan tidak perlu memikirkan dan melakukan perhitungan dari sisi keseimbangan dimana itu sangat rumit. Kinerja dari *mobile robot* bergantung pada komponen-komponen dasar seperti *hardware* dari robot (*body*, motor, sensor, dan lain-lain), kondisi lingkungan tempat robot bekerja, dan *programming* [1].

### 2.2 Konsep Obstacle Avoidance

Dari namanya jelaslah bahwa robot *obstacle avoidance* bertujuan untuk menghindari tabrakan dengan penghalang. Disebut *obstacle avoidance* karena robot ini hanya akan bergerak menghindari objek yang bergerak mendekat di depannya tanpa kembali ke jalurnya.



Gambar 2.1 Konsep Obsbtacle Avoidance

### 2.3 Komponen Robot

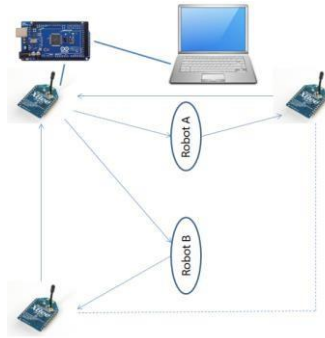
Sensor yang digunakan dalam penelitian ini ada dua jenis, yaitu sensor jarak menggunakan HC SR04 dan sensor kompas menggunakan sensor CMPS 10. Sensor HC SR04 hanya digunakan pada robot A di bagian depan, sedangkan CMPS 10 digunakan pada robot A dan B diletakkan di bagian tengah *chassis*. Untuk mikrokontroler pada robot digunakan Arduino Uno. Arduino UNO adalah papan mikrokontroler berdasarkan ATmega328. Arduino ini berisi semua yang diperlukan untuk mendukung mikrokontroler, untuk mengaktifkan cukup menghubungkannya ke computer dengan kabel USB dengan adaptor AC-DC atau baterai [3]. Kemudahan penggunaan Arduino Uno merupakan alasan dipilihnya komponen tersebut. Driver motor pada penelitian ini menggunakan MOSFET. Untuk display parameter pada robot digunakan LCD 16x2. Sedangkan untuk modul komunikasi digunakan modul *wireless* XBee seri 1 yang bekerja frekuensi radio 2.4 GHz dan *transmit power output* 1mW, jarak yang terjangkaupun sampai 30m untuk *indoor* dan 100m untuk *outdoor*.

### 2.4 Kontrol Proporsional

Kontrol proporsional bekerja dengan cara mengalikan nilai *gain proportional* dengan error. Kelebihan yang dimiliki oleh kontrol proporsional adalah bentuknya yang sederhana dan kecepatan responnya, dimana dalam penelitian ini sangat dibutuhkan kecepatan respon yang baik. Kontrol digunakan untuk mengatur keluaran berupa PWM. *Pulse Width Modulation* (PWM) adalah teknik yang umumnya digunakan untuk mengontrol daya ke perangkat listrik, dirancang praktis dengan *modern switch* daya elektronik. PWM digunakan untuk mendapatkan hasil analog dengan cara digital atau PWM merupakan cara digital tingkat pengkodean sinyal analog[4].

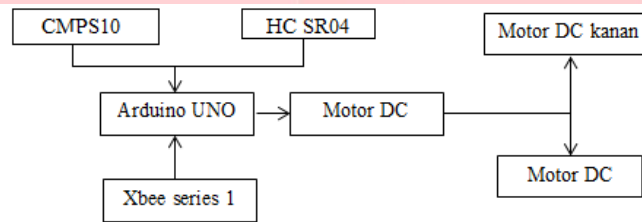
### 2.5 Gambaran Umum Sistem

untuk arena uji coba akan dibuat sebuah *track* maya lurus untuk menjadi lajur jalan robot, yang terletak pada suatu bidang arena berbentuk persegi dengan modul *wireless* XBee Pro di tiap titik sudutnya terhubung dengan satu mikrokontroler yang sama yaitu Arduino Mega. Secara terpisah dari arena akan ada sebuah komputer sebagai pemantau dan pemberi perintah terhadap robot. Seperti yang pada gambar 2.3 yang menunjukkan skema arena uji coba robot beserta alur komunikasinya.



Gambar 2.3 Skema Arena Uji Coba Robot & Alur Komunikasi

Selain percobaan secara sistem menyeluruh sistem internal pada robot pun harus dibangun dengan baik, berikut gambar 2.4 yang menunjukkan sistem internal robot.



Gambar 2.4 Sistem Internal Robot

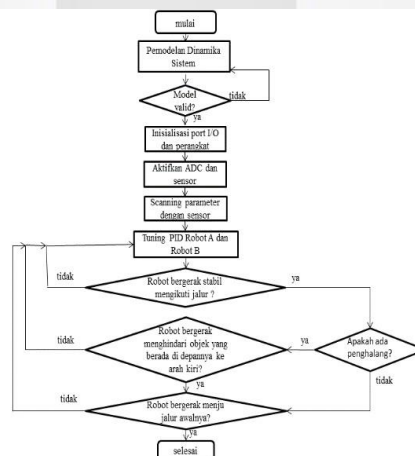
**2.6 Perancangan Perangkat Keras**

Seperti yang dapat dilihat pada gambar 2.4 dan yang sudah dijelaskan pada bagian sebelumnya bahwa robot akan diisi dengan komponen penyusun utama yaitu Motor DC, Driver Motor, Arduino Uno, Xbee series 1, CMPS 10 dan HC SR04. Ban sebagai penggerak akan digunakan tiga buah ban, dua *differential wheel* dan satu *omni wheel*. LCD sebagai penampil nilai PWM kanan dan kiri, juga nilai jarak yang dibaca oleh HC SR04.

Dalam perancangan XBee digunakan 3 macam node yaitu node *Coordinator*, *Router*, dan *End Device*. *Node Coordinator* diperankan oleh XBee yang berada dekat dengan Arduino Mega, *node Router* diperankan oleh dua XBee yang terletak di sudut-sudut arena. Dan *End Device* diperankan oleh XBee pada robot A dan B.

**2.7 Perancangan Perangkat Lunak**

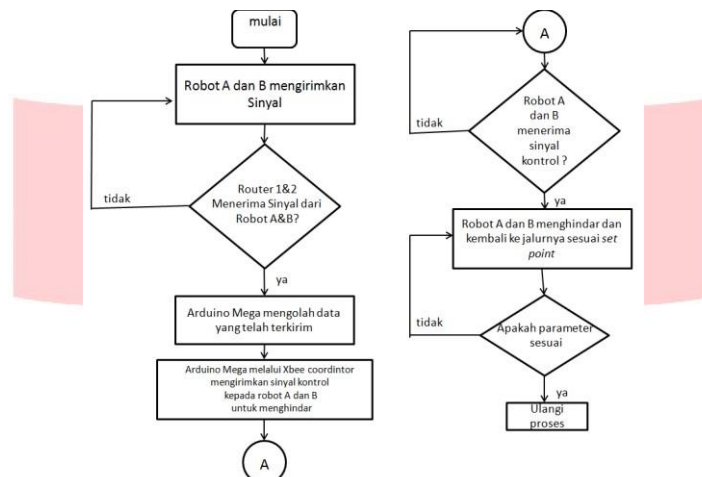
Diagram alir pada gambar 2.5 berikut ini menunjukkan diagram alir perancangan perangkat lunak mobile robot skema collision avoidance untuk robot A dan B.



Gambar 2.5 Diagram Alir Perancangan Perangkat Lunak Mobile Robot Skema Collision Avoidance Untuk Robot A dan B

**2.8 Perancangan Sistem Komunikasi antar mobile robot**

Dalam percobaan ini menggunakan 5 buah perangkat komunikasi Xbee untuk proses komunikasi, dimana dua buah perangkat Xbee dipasang pada masing-masing *mobile robot* dan 2 buah perangkat Xbee (*router*) terhubung langsung di sudut ruangan yang digunakan sebagai penerima data dan penghubung antara 2 *mobile robot* serta 1 buah perangkat Xbee sebagai *coordinator*. Komunikasi yang dirancang pada sistem *mobile robot* ini adalah komunikasi dua arah, namun antara robot A dan robot B tidak berkomunikasi secara langsung melainkan melalui *coordinator* (Arduino Mega). Arduino Mega digunakan sebagai pusat pengontrolan dari sistem yang telah dirancang. Untuk format data kiriman dari Arduino Mega kepada robot A berupa kiriman data PWM (0-255). Tujuannya adalah agar data kiriman cepat sampai. Perancangan komunikasi digambarkan pada gambar 2.7.



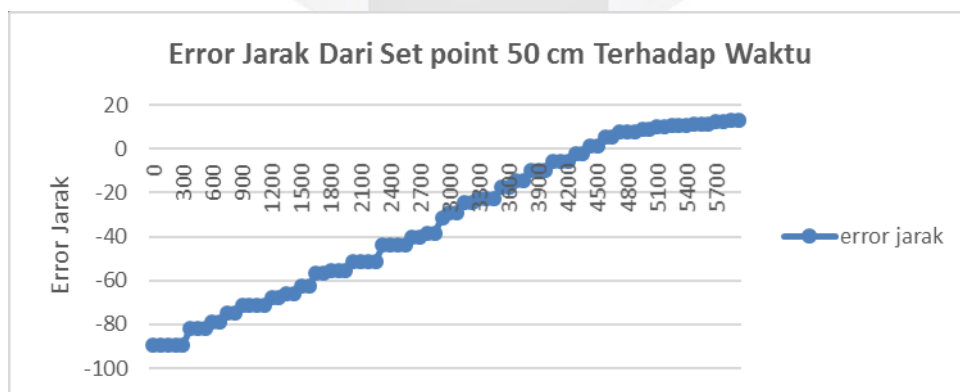
**Gambar 2.6** Diagram Alir Sistem Komunikasi *Collision Avoidance*

Data yang dikirim oleh robot A disusun berupa paket data yang berisi header, suduthigh, sudutlow, sudut\_koma, jarakhigh, jaraklow, dan jarak\_koma. Header yang merupakan penanda untuk setiap *router* yang bertujuan agar paket data yang dikirim untuk A dan B dapat dikenali oleh pengontrol. Dalam perancangan komunikasi ini, untuk robot B header yang digunakan yaitu 'B' sedangkan untuk robot A 'A'. Setelah header parameter yang dikirim selanjutnya berupa nilai sudut. Pemecahan nilai sudut menjadi suduthigh dan sudutlow. Hal ini bertujuan untuk mengatasi ketidakmampuan Xbee dalam mengirim data lebih dari 10 bit sedangkan rentang pembacaan sudut adalah 0-359.9 (9 bit) sehingga pemecahan nilai sudut dibagi menjadi dua bagian. Sementara parameter sudut\_koma yang dikirim dalam bentuk integer sehingga nilai sudut juga harus dipisah dan menjadi bagian tersendiri. Sama halnya dengan nilai sudut, pada penentuan jarak pun nilai jarak dibagi menjadi dua yaitu jarakhigh dan jaraklow karena rentang pembacaan jarak adalah 0 – 400m (9 bit). Sedangkan untuk jarak\_koma perlakuannya sama seperti sudut\_koma.

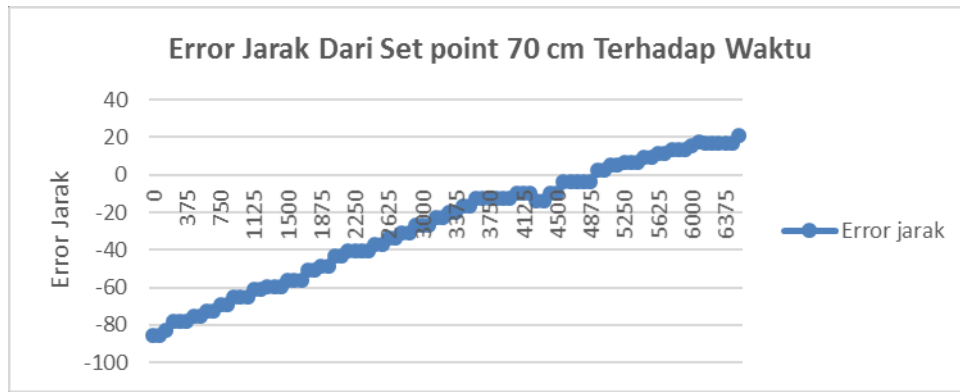
**3. PENGUJIAN DAN ANALISIS**

**3.1 Hasil Pengukuran dan Analisa Jarak**

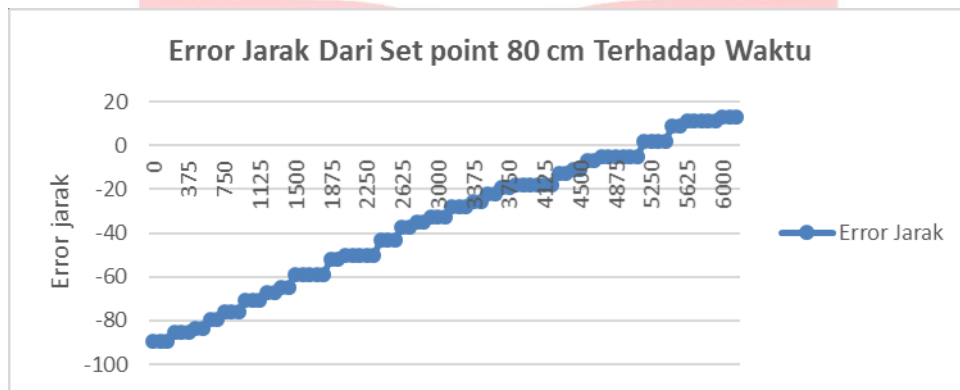
Telah dijelaskan sebelumnya di bagian rumusan masalah, bahwa S adalah jarak yang dibentuk oleh robot saat menghindari, dari percobaan yang sudah dilakukan, didapatkan hasil sebagai berikut.



**Gambar 3.1** Grafik Error Jarak Dari *Set point* 50 cm Terhadap Waktu



Gambar 3.2 Grafik Error Jarak Dari Set point 70 cm Terhadap Waktu



Gambar 3.3 Grafik Error Jarak Dari Set point 80 cm Terhadap Waktu

### 3.2 Pemodelan Sistem

Ada 3 parameter yang perlu dicari nilainya untuk menentukan model dinamika sistem dari motor DC yaitu, momen inersia motor ( $I_m$ ), Torsi maksimum ( $T$ ), dan juga kecepatan sudut tanpa beban ( $\omega_{no-load}$ ).

Nilai momen inersia ( $I$ ) didapatkan dengan mengukur terlebih dahulu massa dan jari-jari roda. Dimana nilai  $J$  sama dengan  $mr^2$ . Perbandingan gear pada roda pun juga harus diketahui besarnya. Lalu berikutnya momen inersia jangkar ( $I_a$ ) dianggap 0 karena nilainya yang sangat kecil. Berikut persamaan momen inersia motor ( $I_m$ ).

$$I = I_a + I_m \tag{3.1}$$

Nilai torsi maksimum, didapatkan dengan cara memberikan PWM maksimal (255) pada motor DC. Lalu motor diikatkan pada dynamo meter untuk menariknya secara vertical, sehingga didapatkan nilai gaya yang dihasilkan. Torsi didapatkan dengan mengalikan gaya dengan jari-jari kelengkungan roda.

Nilai kecepatan sudut tanpa beban ( $\omega_{no-load}$ ) didapatkan dengan cara mengkonversi nilai rpm yang didapat dari mengukur kecepatan roda menggunakan tachometer ke satuan rad/s.

Tabel 3.1 Data Pengukuran Kecepatan Sudut Tanpa Beban

Data ke-	Ea (V)	RPM	$\omega_{no-load}$	Kb(V.s/rad)
1	10.37	1167	122.2079542	0.08485536
2	10.37	1168	122.312674	0.08478271
3	10.37	1163	121.7890752	0.08514721
4	10.37	1167	122.2079542	0.08485536
5	10.37	1168	122.312674	0.08478271
6	10.37	1159	121.3701962	0.085441075
7	10.37	1162	121.6843554	0.085220487
8	10.37	1161	121.5796357	0.085293889
9	10.37	1278	133.831847	0.077485294
10	10.37	1158	121.2654764	0.085514858
Average				0.084337895

Tabel 3.2 Pengukuran Momen Inersia Motor

Massa Roda (kg)	Jari-jari Roda	N1	N2	$J_m$ (kg.m <sup>2</sup> )
0.006	0.02	8	64	0.00000198

Tabel 3.3 Pengukuran Torsi Maksimal Motor

Massa (Kg)	Percepatan Gravitasi	Gaya (N)	r(m)	Torsi (N.m)	Ea (V)	Kt/Ra
0.75	9.8	7.35	0.2	1.47	10.37	0.141755

Pemodelan dinamika sistem dari motor DC dapat dicari dengan persamaan berikut.

$$\frac{\theta_m(s)}{E_a(s)} = \frac{\frac{Kt}{Ra \times Jm}}{s[s + \frac{1}{Jm} (D + Kt \times kb)]} \tag{3.2}$$

Dengan Kb sama dengan Ea/ ω<sub>no-load</sub> dan kt/Ra sama dengan T<sub>stall</sub>/Ea. Setelah parameter-parameter pada persamaan di atas sudah dimasukkan maka akan didapatkan sebuah persamaan seperti berikut

$$\frac{\theta_m(s)}{E_a(s)} = \frac{716000}{s^2 + 60400s} \tag{3.3}$$

Untuk mendapatkan bentuk persamaan dalam kecepatan sudut (ω<sub>m</sub>(s)) maka persamaan diatas harus didiferensiasi dengan cara dikalikan dengan domain S, sehingga persamaannya menjadi seperti di bawah ini.

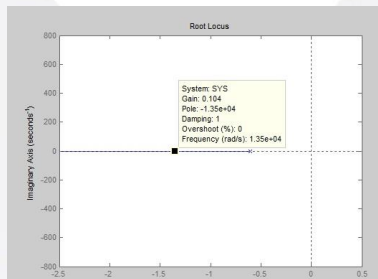
$$\frac{\omega_m(s)}{E_a(s)} = \frac{716000}{s + 60400} \tag{3.4}$$

### 3.4 Implementasi Kontrol

Tahap perancangan kontrol dilakukan dengan metode root locus. Pertama beri input step bernilai satu dan hasilnya adalah

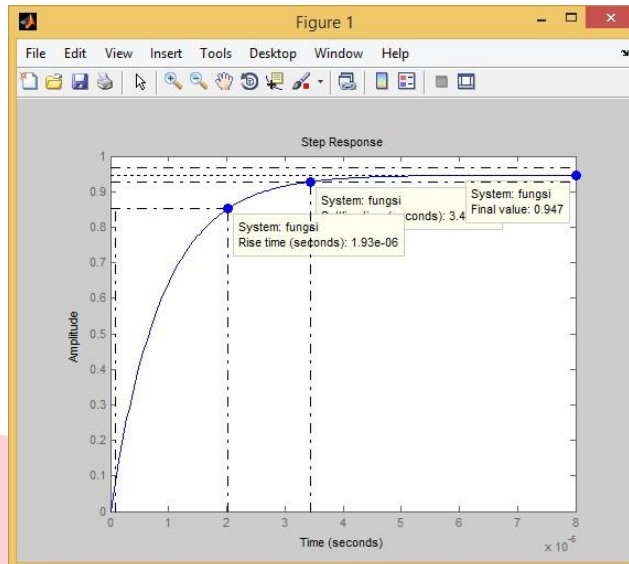
Tabel 3.4 Karakteristik Fungsi Transfer

Tr(s)	Ts(s)	OS(%)	Peak Amp	Nilai Tunak
2.83x10 <sup>-6</sup>	5.04 x 10 <sup>-6</sup>	0	0.922	0.922



Gambar 3.4 Root Locus Fungsi Transfer

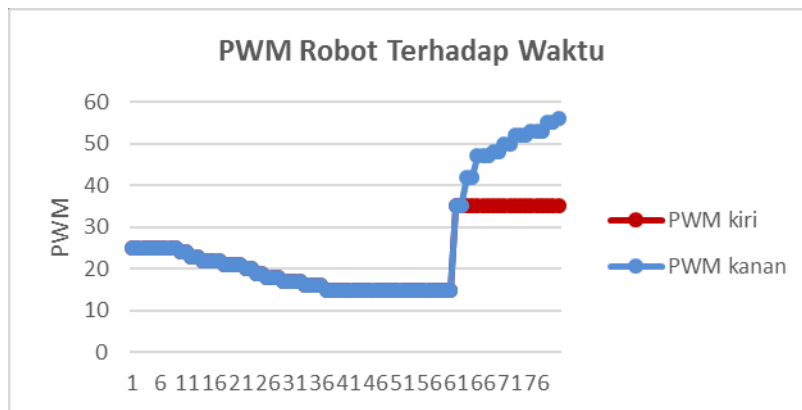
Saat diimplementasikan didapatkan nilai K<sub>p</sub> terbaik adalah 1.5048. Berikut ini adalah hasil simulasinya di matlab setelah diberi gain *proportional*.



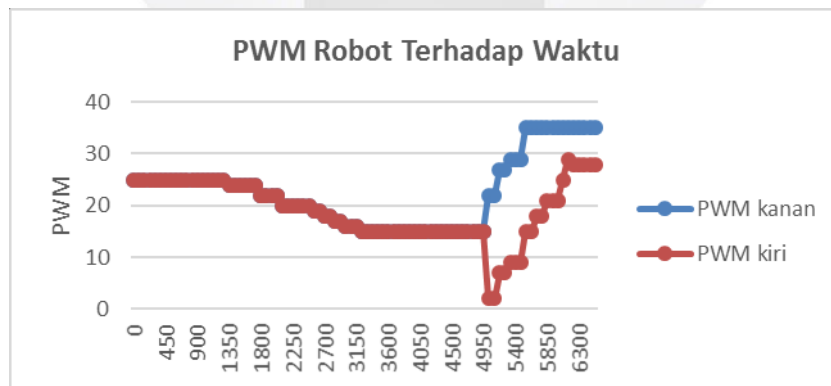
Gambar 3.5 Respon Transien Dari Fungsi Transfer Setelah Diberi  $K_p = 1.5048$

Dapat dilihat pada gambar bahwa nilai respon transien menjadi lebih cepat dengan *rise time*  $1.93 \times 10^{-6}$ , *settling time*  $3.44 \times 10^{-6}$ , sedangkan *peak amplitude* dan *final value*nya menjadi 0.947.

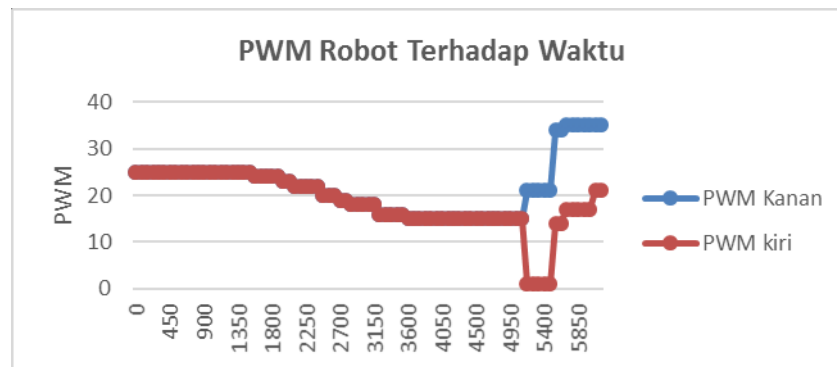
Sedangkan untuk hasil pengontrolan PWM berikut ini grafik dari *set point* 50 cm, 70 cm, dan 80 cm.



Gambar 3.6 Grafik Nilai PWM Kanan dan Kiri Saat Set point Jarak 50 cm



Gambar 3.1 Grafik Nilai PWM Kanan dan Kiri Saat Set point Jarak 70 cm



Gambar 3.8 Grafik Nilai PWM Kanan dan Kiri Saat Set point Jarak 80 cm

#### 4. Kesimpulan & Saran

##### 4.1 Kesimpulan

1. Didapatkan fungsi transfer robot berada pada orde 1 dengan persamaan sebagai berikut  

$$\frac{\omega_m(s)}{E(s)} = \frac{716000}{s+60400}$$
2. Setelah melakukan metode pengontrolan menggunakan *root locus* didapatkan nilai pengontrolan yang paling baik adalah dengan menambahkan nilai  $K_p$  sebesar 1.5048.
3. Penggunaan RSSI dalam penelitian tugas akhir ini tidak efektif karena nilai pengukuran yang tidak stabil.
4. Ditinjau dari respon transiennya setelah pengontrolan, didapatkan nilai *rise time* berubah dari  $2.83 \times 10^{-6}$  menjadi  $1.93 \times 10^{-6}$ , *settling time* dari  $5.04 \times 10^{-6}$  menjadi  $3.44 \times 10^{-6}$ , dan *final value* dari 0.922 menjadi 0.947.
5. Ditinjau dari error jarak yang terjadi di percobaan set point 50 cm, 70 cm, dan 80 cm didapatkan nilainya adalah 1.27 cm, 2.32cm, dan 1.79 cm secara berurutan. Terjadinya nilai error diakibatkan oleh adanya *delay* pengiriman data sebesar 225 mS per satu kali *data sampling*.

##### 4.2 Saran

1. Gunakan lebih banyak sensor jarak HC SR04 untuk *sensing* jarak di area sekitar robot yang lebih baik.
2. Jika ingin membuat skema *obstacle avoidance*, gunakanlah sistem pengontrolan yang *embedded*, agar tidak terganggu oleh *delay* waktu.
3. Penggunaan metode RSSI tidak disarankan dalam proses *positioning* karena data hasil pembacaan Xbee dalam hal *positioning* kurang stabil, sedangkan dalam penelitian ini diperlukan nilai keakuratan data *positioning* yang tepat agar dapat menjalankan skema *obstacle avoidance* yang baik.
4. Gunakan aspek momentum untuk meninjau dan mengaplikasikan skema *obstacle avoidance* dengan menerapkan hukum kekekalan momentum.

##### Daftar Pustaka:

- [1] Reza Fauzi Iskandar, Edi Leksono, and Suprijanto. (2010). *Implementasi Kontrol Adaptif dengan Metode Inverse Dynamic pada Penjejak Lintasan Fleksibel Mobile robot*. Thesis ITB, Bandung.
- [2] Skrjanc, Igor. Gregor Klancar. (2010). *Optimal Cooperative Collision avoidance Between Multiple Robots Based on Bernstein-Bezier Curves*. University of Ljubljana. Online Journal.
- [3] \_\_\_\_\_. "Arduino UNO". Tersedia : [online]. [http://arduino.or.id/hardware/detail\\_hardware/8](http://arduino.or.id/hardware/detail_hardware/8). Diakses pada Maret 25 2014.
- [4] \_\_\_\_\_. "Pulse Width Modulation". Tersedia : [online]. [http://digilib.itelkom.ac.id/index.php?option=com\\_content&view=article&id=1045:pulse-width-modulation-pwm&catid=16:mikroprocessorkontroller&Itemid=14](http://digilib.itelkom.ac.id/index.php?option=com_content&view=article&id=1045:pulse-width-modulation-pwm&catid=16:mikroprocessorkontroller&Itemid=14). Diakses pada Maret 26. 2014.