



**UNIVERSITAS INDONESIA**

**PEMODELAN TRUCK-TRAILER DENGAN PENDEKATAN ORDE  
SATU DAN PENGUJIAN PERANGKATNYA BERBASIS  
MIKROKONTROLER**

**TESIS**

**Diajkan sebagai salah satu syarat untuk memperoleh gelar Magister Teknik**

**ENDA WISTA SINURAYA**

**0706173225**

**PROGRAM STUDI TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS INDONESIA  
JULI 2010**



## HALAMAN PERNYATAAN ORISINALITAS

Tesis ini adalah hasil karya saya sendiri, dan semua sumber baik yang diikuti maupun dirujuk telah saya nyatakan dengan benar.

Nama : ENDA WISTA SINURAYA

NPM : 0706173225

Tanda tangan :



Tanggal : 15 Juli 2010

## PENGESAHAN DEWAN PENGUJI

Thesis ini diajukan oleh :  
Nama : Enda Wista Sinuraya  
NPM : 0706173225  
Program studi : Teknik Elektro  
Judul : Pemodelan Truck-Trailer dengan Pendekatan  
Orde satu dan Simulasi Implementasinya dengan  
Line Follower Berbasis Mikrokontroler

Telah berhasil dipertahankan di hadapan Dewan Penguji dan di terima sebagai bagian persyaratan yang di perlukan untuk memperoleh gelar Magister Teknik pada Program Studi Teknik Elektro Fakultas Teknik Universitas Indonesia

### DEWAN PENGUJI

1. Dr. Abdul Muis, ST, M.Eng. (.....)
2. Dr. Abdul Halim, M.Eng. (.....)
3. Dr.Ir.Feri Yusivar M.Eng. (.....)
4. Ir. Aries Subiantoro, M.SEE. (.....)

Ditetapkan di : Depok

Tanggal : 15 Juli 2010

## UCAPAN TERIMAKASIH

Puji sukur, atas segala karunia dan petunjuk dari Allah SWT. Karena kasih-Nya sehingga penulis dapat menyelesaikan penyusunan laporan tesis ini. Penyusunan laporan tesis ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Magister Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Untuk itu penulis mengucapkan terima kasih yang sangat mendalam kepada :

1. Bapak saya Ngukurken sinuraya B.S.c dan ibu saya Debora Elfina br giting yang telah mendoakan saya dan memberikan semangat.
2. Bapak Dr. Abdul Muis S.T., M.Eng, dan Bapak Dr. Abdul Halim, M.Eng., selaku dosen pembimbing yang telah menyediakan waktu, tenaga dan pikiran didalam mengarahkan penulis selama penelitian dan penyusunan laporan tesis ini.
3. Bapak Dr.Ir.Feri Yusivar M.Eng, dan Bapak Ir. Aries Subiantoro, M.SEE., sebagai dosen penguji.
4. Rekan-rekan mahasiswa departemen elektro khususnya program study control industri.

Dan semoga tesis ini membawa manfaat bagi pengembangan ilmu.

Depok, 15 Juli 2010

Penulis



**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Enda Wista Sinuraya  
NPM : 0706173225  
Program Studi : Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis karya : Tesis

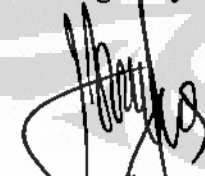
demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

**Pemodelan Truck-Trailer dengan Pendekatan Orde satu dan  
Pengujian Perangkatnya Berbasis Mikrokontroler**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelolanya dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada tanggal : Juli 2010  
Yang menyatakan



( Enda Wista Sinuraya )

## ABSTRAK

Nama : Enda Wista Sinuraya  
Program studi : Teknik Elektro  
Judul : Pemodelan Truck-Trailer dengan Pendekatan Orde satu dan Pengujian Perangkatnya Berbasis Mikrokontroler

Dalam tesis ini dibahas mengenai system truck dengan tiga trailer dengan pendekatan orde 1. Model dianalisa dalam gerak maju dengan pembatasan-pembatasan kecepatan linier dan besar sudut kemudi untuk menghindari fenomena *Jack-Knife*. Pengendali yang dirancang adalah pengendali proportional. Dalam tesis ini juga dibahas tentang perancangan prototipe truck-trailer meliputi perangkat keras dan perangkat lunak. Pengujian dan simulasi masing-masing komponen perangkat sistem truck-trailer berbasis mikrokontroler ATMEGA 8535. Pengujian meliputi pengujian sensor, pengujian aktuator dan pengujian integrasi antara sensor dan aktuator dan komunikasi serial antara ATMEGA 8535 dengan PC

Kata kunci : *Mobile Robot, Trailer, sudut kemudi, AVR ATMEGA 8535, PC*

## ABSTRACT

Name : Enda Wista Sinuraya

Study Program: Electrical Engineering

Title : Truck Trailer Modelling by using first order Approach and testing each component with Microcontroller Based.

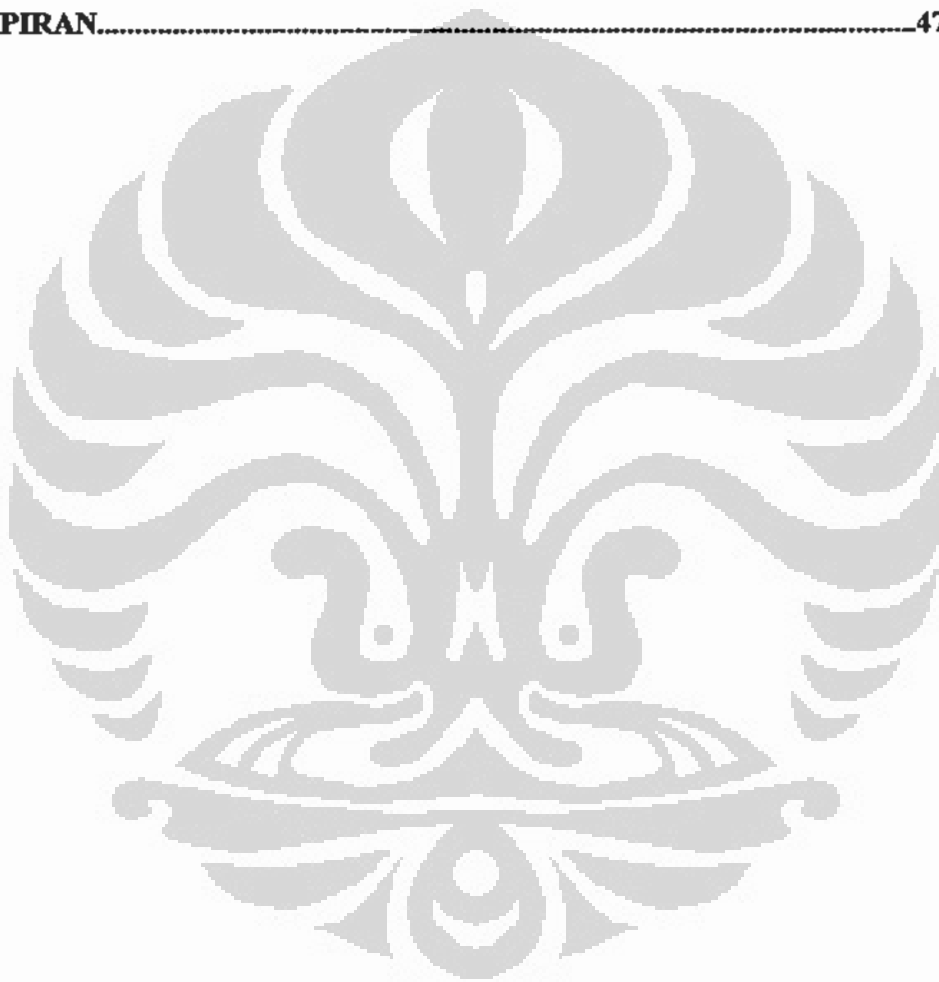
This thesis describes modelling of first order model derived from non-linear truck-trailer system. The model is analyzed in forward direction with applying constrain velocity and steering angle in order to avoid jack-knife phenomena. The proportional type controller is applied to the model. Construction of prototype of the truck-trailer system is described include architecture of hardware and software. Simulation and testing each component of prototype truck-trailer based on ATMEGA 8535. The testing involves testing of sensors, actuators, integration sensors and actuators and serial communication between ATMEGA 8535 and PC.

Key words: Mobile Robot, Trailer, steering angle, AVR Atmega8535, PC

## DAFTAR ISI

<b>HALAMAN PERNYATAAN ORISINALITAS.....</b>	<b>ii</b>
<b>LEMBAR PENGESAHAN.....</b>	<b>iii</b>
<b>UCAPAN TERIMA KASIH.....</b>	<b>iv</b>
<b>HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH .....</b>	<b>v</b>
<b>ABSTRAK.....</b>	<b>vi</b>
<b>ABSTRACT .....</b>	<b>vii</b>
<b>DAFTAR ISI.....</b>	<b>viii</b>
<b>DAFTAR GAMBAR.....</b>	<b>x</b>
<b>DAFTAR TABEL.....</b>	<b>xi</b>
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan Penulisan.....	2
1.3 Pembatasan Masalah .....	2
1.4 Sistematika Penulisan.....	2
<b>BAB 2 PEMODELAN DAN SIMULASI MOBILE ROBOT-3 TRAILERS .....</b>	<b>3</b>
2.2 Model Kinematik Mobile Robot .....	3
2.2 Model Kinematis Mobile Robot-3 Trailers.....	8
2.3 Kendali proportional mobile robot.....	10
2.4 Implementasi Kendali Proportional pada mobile robot-3 trailers.....	12
<b>BAB 3 PERANCANGAN PERANGKAT KERAS DAN PERANGKAT LUNAK MOBILE ROBOT -3 TRAILERS.....</b>	<b>18</b>
3.1 Perancangan Perangkat Keras .....	18
3.1.1 Konstruksi Rangka.....	18
3.1.2 Aktuator.....	19
3.1.3 Sensor.....	24
3.2 Perancangan Perangkat Lunak .....	28
3.2.2 Program Mikrokontroler ATMEGA 8535 .....	28
3.2.3 Komunikasi PC dengan Mikrokontroler .....	32

<b>BAB 4 SIMULASI DAN UJI COBA KOMPONEN</b> .....	<b>33</b>
4.1 Simulasi dan Pengujian Motor Servo untuk Kemudi Truck .....	33
4.2 Simulasi Motor DC dengan Feedback Encoder .....	34
4.3 Simulasi dan Pengujian Sensor Photorelector .....	38
4.4 Pengujian Sensor Sudut Trailer dengan Potensiometer.....	40
<b>BAB 5 KESIMPULAN</b> .....	<b>44</b>
<b>DAFTAR REFERENSI</b> .....	<b>45</b>
<b>DAFTAR ACUAN</b> .....	<b>46</b>
<b>LAMPIRAN</b> .....	<b>47</b>

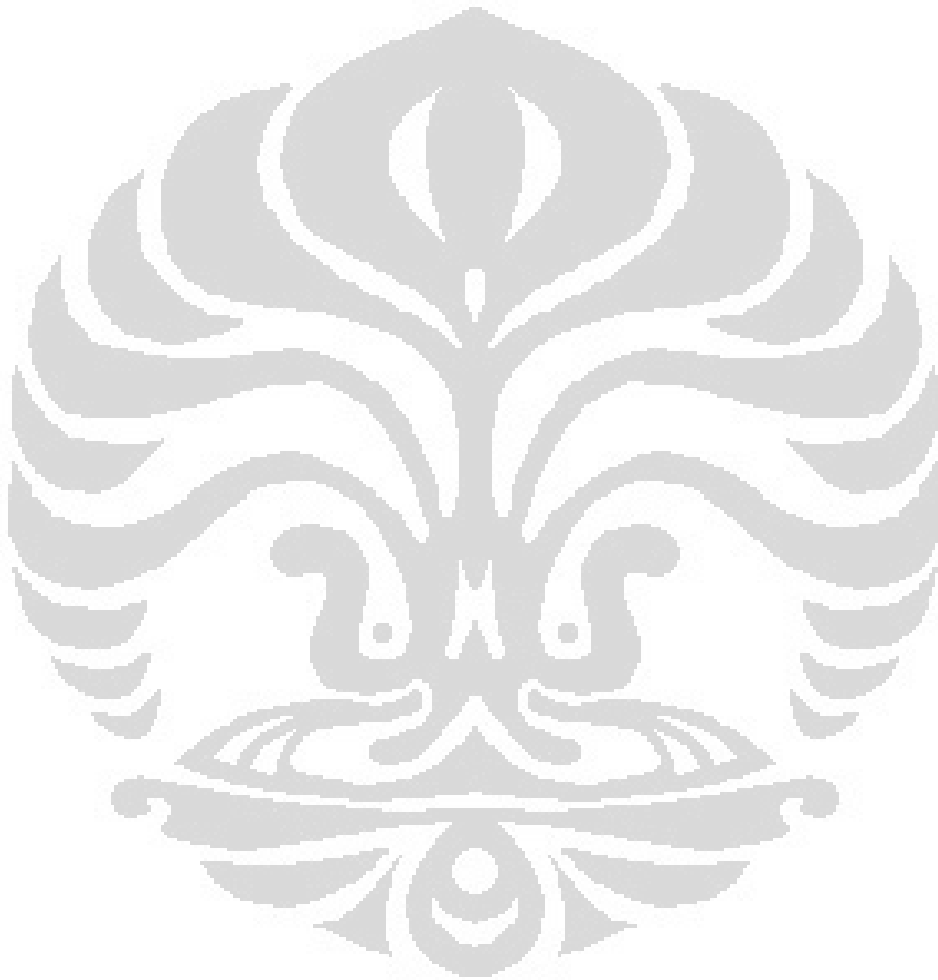


## DAFTAR GAMBAR

Gambar 2.1. Navigasi kemudi mobile robot dalam kawasan 2D.	3
Gambar 2.2. Koordinat sistem mobile robot	4
Gambar 2.3. Koordinat lintasan mobile robot	8
Gambar 2.4. Mobile Robot-3 Trailer	9
Gambar 2.5. Simulasi Mobile Robot	12
Gambar 2.6. Step Respon Sistem Mobile Robot	12
Gambar 2.7. Simulasi Mobile Robot-3 Trailers	13
Gambar 2.8. Sudut Simulasi Mobile Robot-3 Trailers	14
Gambar 2.9. Posisi XY Simulasi Mobile Robot-3 Trailers	17
Gambar 3.1. Mobile Robot-3 Trailer	18
Gambar 3.2. Konstruksi Fisik Motor Servo	19
Gambar 3.3. Posisi Motor Servo	21
Gambar 3.4. Susunan kabel servo motor	21
Gambar 3.5. Rangkaian Driver motor DC dengan L298	23
Gambar 3.6. Ilustrasi Timing enable pada IC	23
Gambar 3.7. Rotary encoder	24
Gambar 3.8. Photoreflector	25
Gambar 3.9. Rangkaian Triger Photoreflektor ke mikrokontroler	26
Gambar 3.10. Rangkaian Kendali Sudut dengan Potensiometer	27
Gambar 3.11. Flowchart Perancangan Perangkat Lunak	28
Gambar 3.12. Pengendalian Motor DC	29
Gambar 3.13. Pengendalian Motor Servo	30
Gambar 3.14. Flowchart Mobile Robot-3 Trailers	31
Gambar 3.15. Komunikasi serial ATMEGA 8535 dengan PC	32
Gambar 4.1. Rangkaian Pengujian Motor Servo	34
Gambar 4.2. Rangkaian Pengujian Motor DC dengan Feedback encoder	37
Gambar 4.3. Rangkaian Pengujian Photoreflector	39
Gambar 4.4. Rangkaian Pengujian Potensiometer	41
Gambar 4.5. Model Mobile Robot-3 Trailers	43

## DAFTAR TABEL

Tabel 2.1 Nilai Radian dan Tangen sudut	6
Tabel 3.1 Spesifikasi teknis motor servo S3003 Futaba	21
Table 4.1 Percobaan Motor Servo	35
Tabel 4.2 Simulasi Motor DC dengan Feedback Encoder	37
Tabel 4.3 Pengujian Photoreflector	40
Tabel 4.4 Pengujian Sensor Sudut Trailer dengan potensiometer	42



## **BAB 1 PENDAHULUAN**

### **1.1 Latar Belakang**

Pemodelan kinematik sistem *Truck-3Trailer* menghasilkan model yang nonlinier. Model ini dapat dilinierisasi menjadi sistem orde 1 dengan memberi batasan (constraint) masukan. Untuk mempercepat proses dibutuhkan sistem kendali. Salah satu jenis pengendali yang paling populer dan telah lama digunakan adalah pengendali *Proportional Integrator Differentiator* (PID). Pengendali PID merupakan pengendali konvensional tetapi masih sangat banyak dijumpai pada dunia industri dan bidang-bidang kendali yang lain. Keuntungan pengendali PID sehingga jenis pengendali ini masih dapat bertahan sampai saat ini adalah proses perancangan dan realisasi pengendali ini tidak terlalu sukar sehingga implementasinya masih dapat berkembang. Dalam tesis ini juga dibahas tentang pengujian dan simulasi masing-masing komponen perangkat sistem truck-trailer berbasis mikrokontroler ATMEGA 8535. Pengujian meliputi pengujian sensor, pengujian aktuator dan pengujian integrasi antara sensor dan aktuator dan komunikasi serial antara ATMEGA 8535 dengan PC.

### **1.2 Tujuan**

Tujuan dari Tesis ini adalah untuk mensimulasikan *Truck-3Trailer* untuk diterapkan dengan mikrokontroler ATMEGA 8535 dengan menguji masing-masing komponen.

### **1.3 Pembatasan Masalah**

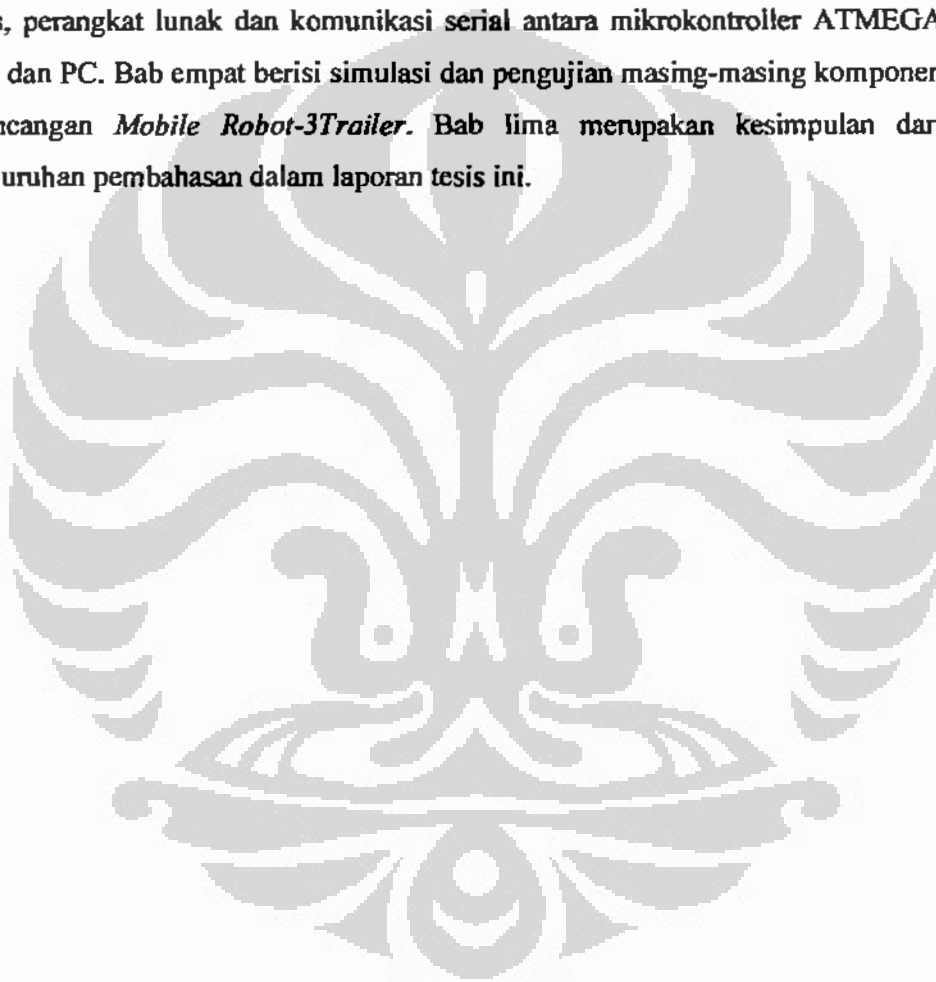
Simulasi dengan program simulasi model *Truck-3Trailer* mengikuti *trajectory* yang diberikan. Pada tesis ini hanya menguji dan mensimulasikan masing-masing komponen perancangannya, belum diterapkan pada sistem secara keseluruhan.



#### 4.4. Sistematika Penulisan

Penulisan thesis ini akan dibagi ke dalam lima bab yang akan menjelaskan secara bertahap mengenai keseluruhan isi thesis ini.

Bab satu merupakan pendahuluan yang berisi latar belakang, tujuan, pembatasan masalah, dan sistematika penulisan. Bab dua membahas dasar teori yaitu pemodelan dan simulasi *Mobile Robot-3Trailer* yang mengikuti *trajectory* yang diberikan dengan program simulasi. Bab tiga membahas mengenai perangkat keras, perangkat lunak dan komunikasi serial antara mikrokontroler ATMEGA 8535 dan PC. Bab empat berisi simulasi dan pengujian masing-masing komponen perancangan *Mobile Robot-3Trailer*. Bab lima merupakan kesimpulan dari keseluruhan pembahasan dalam laporan tesis ini.



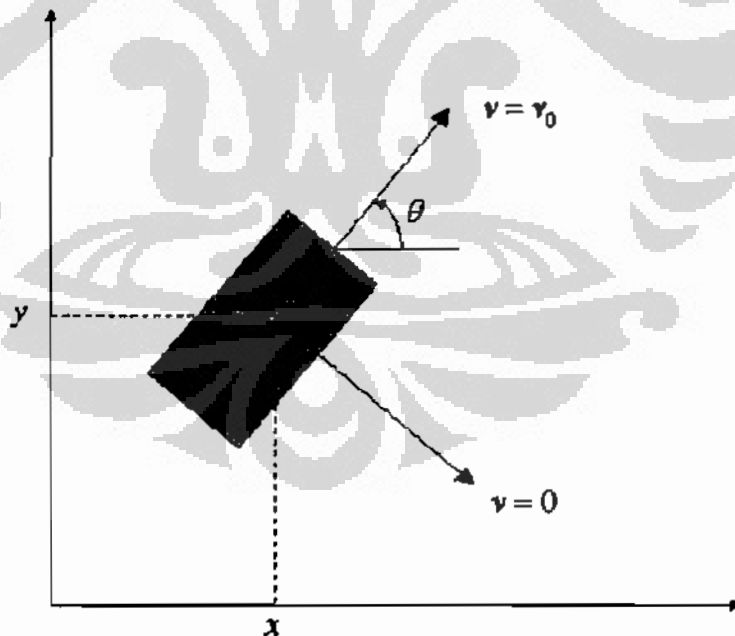
## BAB 2 PEMODELAN DAN SIMULASI MOBILE ROBOT -3 TRAILERS

Bab ini akan membahas model kinematik poros roda depan dan poros roda belakang mobile robot dan model kinematik mobile robot dengan 3 buah *trailer*.

### 3.2 Model Kinematik Mobile Robot

Salah satu contoh robot aplikasi yang memiliki struktur kinematik *nonholonomic* adalah mobile robot. Mobile robot didefinisikan bergerak dalam kawasan 2D. Kontur medan yang tidak rata tidak dimasukkan sebagai unsur Z (sumbu tegak lurus) karena navigasi (gerak robot) tetap bisa diasumsikan bergerak dalam kawasan sumbu XY saja.

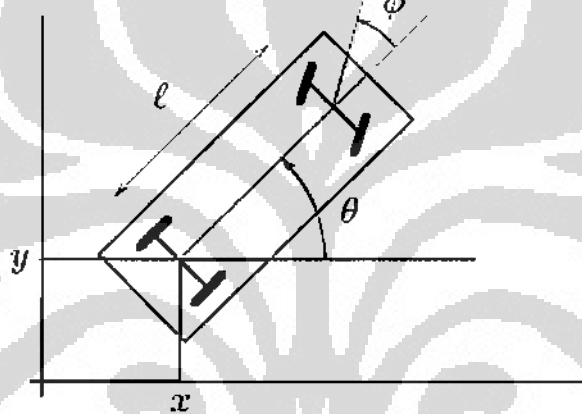
Model yang digunakan dalam tulisan ini adalah model kinematik, oleh sebab itu unsur-unsur dinamik mobile robot seperti berat, gesekan, gravitasi, *slip*, gaya *centrifugal* tidak digunakan dalam pemodel sistem.



Gambar 2.1 Navigasi kemudi mobile robot dalam kawasan 2D.

Gambar 2.1 menggambarkan *nonholonomic constraints* dari navigasi kemudi mobile robot. Dari gambar 2.1 dapat dilihat bahwa tidak ada pergerakan mobile robot pada arah sumbu tegak lurus.

Posisi dan orientasi mobile robot pada sistem sumbu koordinat dipengaruhi oleh 3 variabel. Gambar 2.2 menunjukkan komponen-komponen variabel tersebut. Pertama adalah Posisi X dan Y yang menunjukkan letak titik tengah poros roda belakang. Kedua adalah  $\theta$  yang menunjukkan besar sudut mobile robot terhadap sumbu X. Dan yang ketiga adalah  $\phi$  menunjukkan besar sudut kemudi mobile robot.



Gambar 2.2 Koordinat sistem mobile robot

Dari gambar 2.2 dapat dilihat bahwa kecepatan mobile robot terhadap sumbu X dan sumbu Y dirumuskan sebagai berikut:

$$\dot{x} = v \cos \theta \quad (2.1)$$

$$\dot{y} = v \sin \theta \quad (2.2)$$

$$v^2 = \dot{x}^2 + \dot{y}^2 \quad (2.3)$$

Dimana  $v$  adalah kecepatan linier mobile robot.  $\dot{x}$  adalah kecepatan linier mobile robot terhadap sumbu X dan  $\dot{y}$  adalah kecepatan linier mobile robot terhadap sumbu Y.

Posisi tengah poros roda depan mobile robot dirumuskan sebagai berikut:

$$x_f = x + l \cos \theta \quad (2.4)$$

$$y_f = y + l \sin \theta \quad (2.5)$$

Dimana  $x$  dan  $y$  adalah posisi tengah poros roda belakang mobile robot.

Kecepatan poros roda depan diturunkan dari posisi poros roda depan pada persamaan (2.3) dan persamaan (2.4), sehingga dapat dirumuskan menjadi:

$$\dot{x}_f = \dot{x} - \dot{\theta} l \sin \theta \quad (2.6)$$

$$\dot{y}_f = \dot{y} + \dot{\theta} l \cos \theta \quad (2.7)$$

*Nonholonomic constraints* untuk poros roda depan dan poros roda belakang adalah:

$$\dot{x}_f \sin(\theta + \phi) - \dot{y}_f \cos(\theta + \phi) = 0 \quad (\text{front wheel}) \quad (2.8)$$

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0 \quad (\text{rear wheel}) \quad (2.9)$$

Substitusi persamaan (2.5) dan (2.6) ke persamaan (2.7) menghasilkan persamaan sebagai berikut:

$$(\dot{x} - \dot{\theta} l \sin \theta) \sin(\theta + \phi) - (\dot{y} + \dot{\theta} l \cos \theta) \cos(\theta + \phi) = 0$$

$$\dot{x} \sin(\theta + \phi) - \dot{\theta} l \sin \theta \sin(\theta + \phi) - \dot{y} \cos(\theta + \phi) - \dot{\theta} l \cos \theta \cos(\theta + \phi) = 0$$

$$\dot{x} \sin(\theta + \phi) - \dot{y} \cos(\theta + \phi) - \dot{\theta} l (\cos \theta \cos(\theta + \phi) + \sin \theta \sin(\theta + \phi)) = 0$$

$$\dot{x} \sin(\theta + \phi) - \dot{y} \cos(\theta + \phi) - \dot{\theta} l (\cos \theta (\cos \theta \cos \phi - \sin \theta \sin \phi) + \sin \theta (\sin \theta \cos \phi + \cos \theta \sin \phi)) = 0$$

$$\dot{x} \sin(\theta + \phi) - \dot{y} \cos(\theta + \phi) - \dot{\theta} l (\cos^2 \theta \cos \phi - \sin \theta \cos \theta \sin \phi + \sin^2 \theta \cos \phi + \sin \theta \cos \theta \sin \phi) = 0$$

$$\dot{x} \sin(\theta + \phi) - \dot{y} \cos(\theta + \phi) - \dot{\theta} l \cos \phi = 0$$

$$\dot{x} \sin \theta \cos \phi + \dot{x} \cos \theta \sin \phi - \dot{y} \cos \theta \cos \phi + \dot{y} \sin \theta \sin \phi - \dot{\theta} l \cos \phi = 0$$

$$\cos \theta (\dot{x} \sin \theta - \dot{y} \cos \theta) + \sin \phi (\dot{x} \sin \theta + \dot{y} \cos \theta) - \dot{\theta} l \cos \phi = 0$$

$$v \sin \phi = \dot{\theta} l \cos \phi \quad (2.10)$$

$$\dot{\theta} = \frac{v}{l} \tan \phi \quad (2.11)$$

Dari persamaan (2.11) masukan sudut kemudi ( $\phi$ ) mempunyai nilai yang tidak linear. Jika sudut kemudi ( $\phi$ ) 90 derajat, nilai  $\theta$  menjadi tidak terhingga. Fenomena ini disebut *jack-knife*.

Tabel 2.1 Nilai Radian dan Tangen sudut

Sudut (derajat)	Radian	Tangen
5	0.0872	0.0874
10	0.1745	0.1763
15	0.2618	0.2679
20	0.3490	0.3639
25	0.4363	0.4663
30	0.5236	0.5773
45	0.785	1
60	1.0466	1.7320
75	1.3083	3.7320
80	1.3955	5.6712
90	1.57	$\infty$

Dari tabel (2.1) dapat dilihat nilai radian sudut kemudi ( $\phi$ ) dengan nilai tangen sudut kemudi ( $\phi$ ). Untuk menghindari fenomena *jack-knife*, pada simulasi

sudut kemudi ( $\theta$ ) dibatasi lebih kecil dari 90 derajat dan lebih besar dari -90 derajat.

Dari hasil simulasi perubahan sudut kemudi ( $\theta$ ) mobile robot sangat kecil terhadap fungsi waktu. sehingga persamaan (2.1) dapat dirumuskan menjadi:

$$\tan \theta = \theta$$

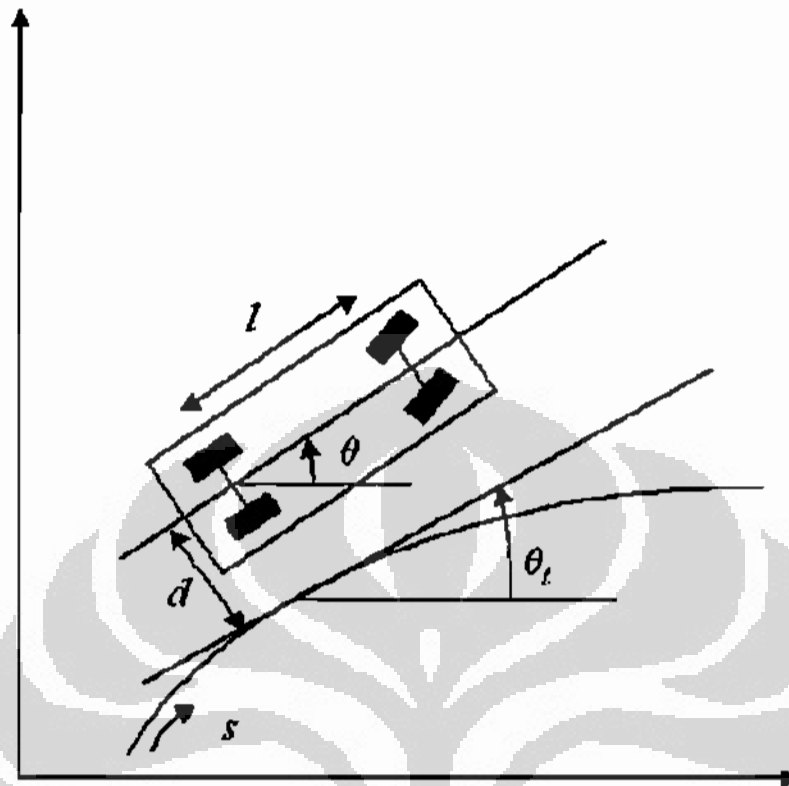
$$\dot{\theta} = \frac{v}{l} \theta \quad (2.12)$$

Matrik model kinematik mobile robot adalah sebagai berikut:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \frac{\tan \theta}{l} \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2 \quad (2.13)$$

$$\dot{q} = g_1 v_1 + g_2 v_2 \quad (2.14)$$

Dimana  $v_1$  adalah kecepatan linear mobile robot dan  $v_2$  adalah kecepatan angular mobile robot.

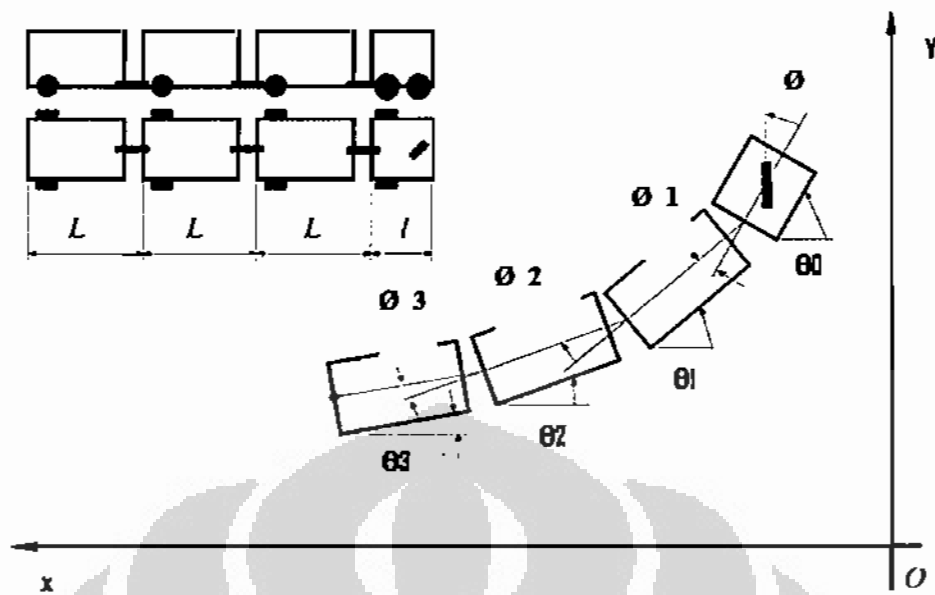


Gambar 2.3 Koordinat lintasan mobile robot

Lintas mobile robot terhadap *trajectory* ditunjukkan gambar 2.3. Besar beda jarak antara posisi poros roda belakang dengan posisi *trajectory* adalah  $d$ . Sudut antara mobile robot dengan sudut *trajectory* adalah  $\theta_p = \theta - \theta_t$ .

### 3.2.1 Model Kinematik mobile robot-3 Trailers

Jenis *trailer* yang digunakan dalam simulasi adalah jenis pasif, ini berarti *trailer* tidak dapat dikendalikan secara *independent*. Oleh sebab itu, untuk mengendalikan pergerakan *trailer* hanya dapat dilakukan dengan mengendalikan pergerakan mobile robot.



Gambar 2.4 Mobile Robot-3 Trailer

Gambar 2.2 menunjukkan sistem mobile robot dengan 3 buah *trailer* bergandengan yang pasif dimana:

- $\theta$  : sudut kemudi mobile robot
- $\theta_0$  : Besar sudut mobile robot terhadap sumbu X
- $\theta_1$  : Besar selisih sudut mobile robot dengan *trailer* 1
- $\theta_1$  : Besar sudut *trailer* 1 terhadap sumbu X
- $\theta_2$  : Besar selisih sudut *trailer* 1 dan *trailer* 2
- $\theta_2$  : Besar sudut *trailer* 2 terhadap sumbu X
- $\theta_3$  : Besar selisih sudut *trailer* 2 dan *trailer* 3
- $\theta_3$  : Besar sudut *trailer* 3 terhadap sumbu X



Posisi X dan Y masing-masing *trailer* dapat dirumuskan sebagai berikut:

$$x_i = x_{i-1} - l_i \cos \theta_i \quad (2.14)$$

$$y_i = y_{i-1} - l_i \sin \theta_i \quad (2.15)$$

Dari persamaan (2.10) dapat dirumuskan besar sudut masing-masing *trailer* terhadap sumbu X sebagai berikut (G.Oriolo-Mobile Robotics: Control Problems):

$$\begin{aligned} \phi &= \theta_i - \theta_{i+1} \\ v_{i+1} &= \cos(\theta_i - \theta_{i+1}) v_i \\ \dot{\theta}_{i+1} &= \frac{v_{i+1} \sin(\theta_i - \theta_{i+1})}{l_{i+1} \cos(\theta_i - \theta_{i+1})} \\ \dot{\theta}_{i+1} &= \frac{v_i}{l_{i+1}} \sin(\theta_i - \theta_{i+1}) \end{aligned} \quad (2.16)$$

Oleh karena mobile robot bergandeng dengan *trailer* maka kecepatan mobile robot dan *trailer* sama, sehingga persamaan (2.16) menjadi:

$$\dot{\theta}_{i+1} = \frac{v}{l_{i+1}} \sin(\theta_i - \theta_{i+1}) \quad (2.17)$$

### 2.3 Kendali Proportional Mobile Robot

PID (Proportional Integral Derivatif) banyak digunakan di pengendalian industri. kelebihan PID:

1. Mudah dalam hal penalaan/tuning
2. Sudah banyak tertanam/fitur dari controller seperti PLC
3. Terbukti handal dioperasikan di pengendalian pabrik
4. Algoritma-algoritma pengendalian lain seperti

*cascade, ratio, feedforward* masih berbasis PID.

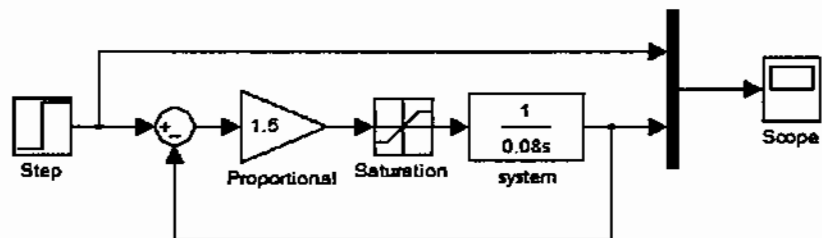
#### 5. Mudah dianalisa

#### Kelemahan PID:

1. Masih single input single output (siso).
2. Karena *variable terkontrol* di plant tidak linear, maka pentuningan hanya bagus didaerah operasional (range operasi sempit).
3. Sulit untuk *process control optimization*.

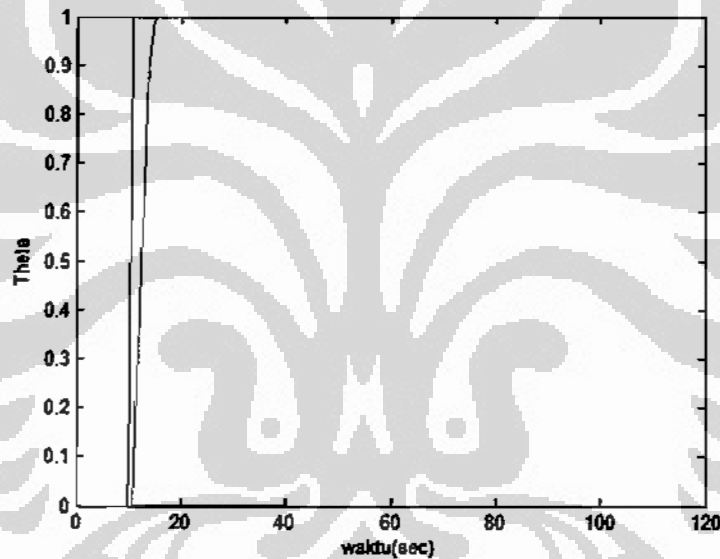
Dari persamaan (2.11) sistem *mobile robot* merupakan sistem orde satu. Pada uji coba ini pengendalian sistem menggunakan kendali *proportional*. Sudut kemudi *mobile robot* dibatasi antara  $0^\circ$  sampai  $15^\circ$ . Pengendali *proportional* digunakan untuk mengendalikan *mobile robot*. Sedangkan *trailer* tidak dapat dikendalikan karena bersifat pasif. *Trajectory mobile robot* yang dirancang mempunyai pergerakan ke arah sumbu X yang konstan terhadap fungsi waktu. Dalam simulasi ini, *Mobile robot* bergandengan dengan 3 buah *trailer*. Hasil akhir simulasi yang hendak dicapai adalah kesalahan minimum ( $\theta_p$ ) sudut *mobile robot* dan *trailer* dengan sudut *trajectory*. Kemudian kesalahan minimum (*error*) posisi *mobile robot* dan masing-masing *trailer* dengan *trajectory*.

Untuk menghitung besarnya nilai penguatan *proportional*, dilakukan simulasi kendali *mobile robot* dengan program simulasi yang dapat dilihat seperti gambar (2.5) dengan panjang truck 0.08 m, panjang trailer 0.13 m dan kecepatan 0.5 m/s .



Gambar 2.5 Simulasi Mobile Robot

Sistem *mobile robot* adalah orde satu. Dari hasil simulasi penguatan 1.5 dan masukan step menghasilkan respon seperti gambar 2.6.



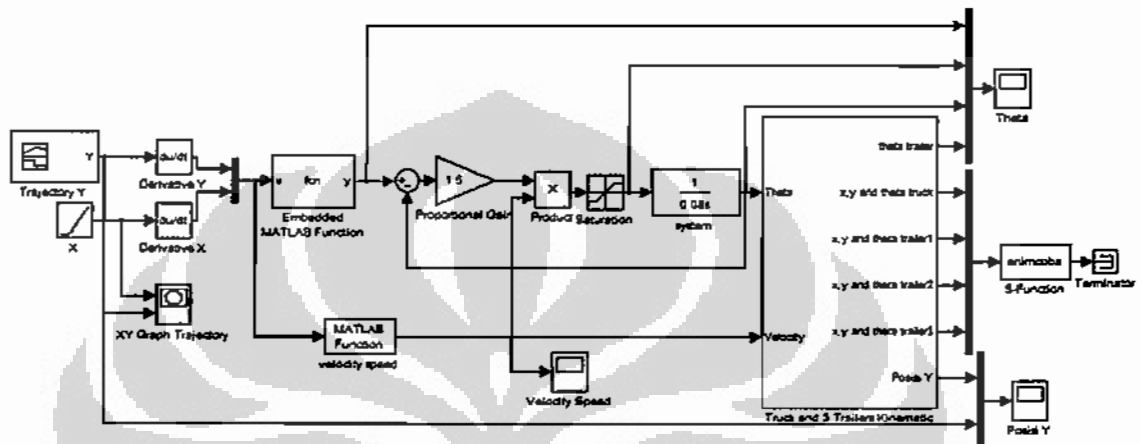
Gambar 2.6 Step Respon Sistem *Mobile Robot*

Dari gambar 4.2 diperoleh nilai delay time sebesar 0.1 detik dan nilai konstanta waktu sistem *mobile robot* 0.29 detik.

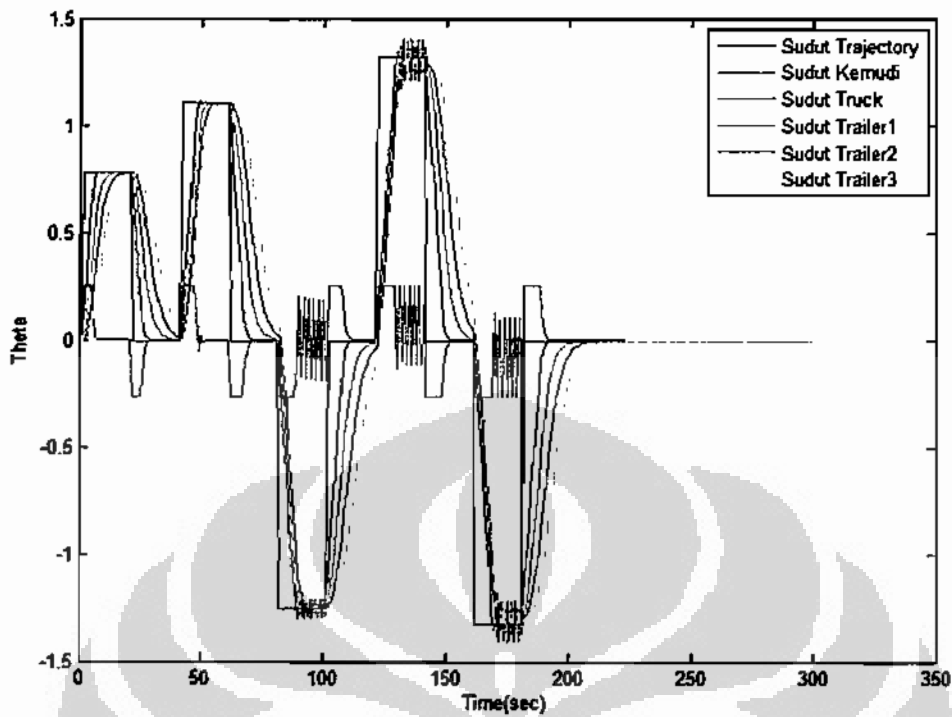
#### 2.4 Implementasi Kendali Proportional pada *mobile robot-3 trailers*

Kendali proportional yang telah dirancang diterapkan pada sistem *mobile robot-3 Trailers* dengan lintasan yang memiliki sudut yang bervariasi dapat dilihat pada gambar 2.7 dibawah ini.

Kecepatan mobile robot merupakan konstanta gain yang dapat dikalikan dengan nilai penguatan proportional. Pada simulasi ini diberikan sebuah blok saturasi untuk dapat mengatur nilai sudut kemudi ( $\theta$ ) yang diinginkan. Saturasi ini juga dapat mencegah terjadinya fenomena *jack-knife* dimana kemudi ( $\theta$ ) 90 derajat, maka dari persamaan (2.11) nilai  $\theta$  menjadi tidak terhingga.

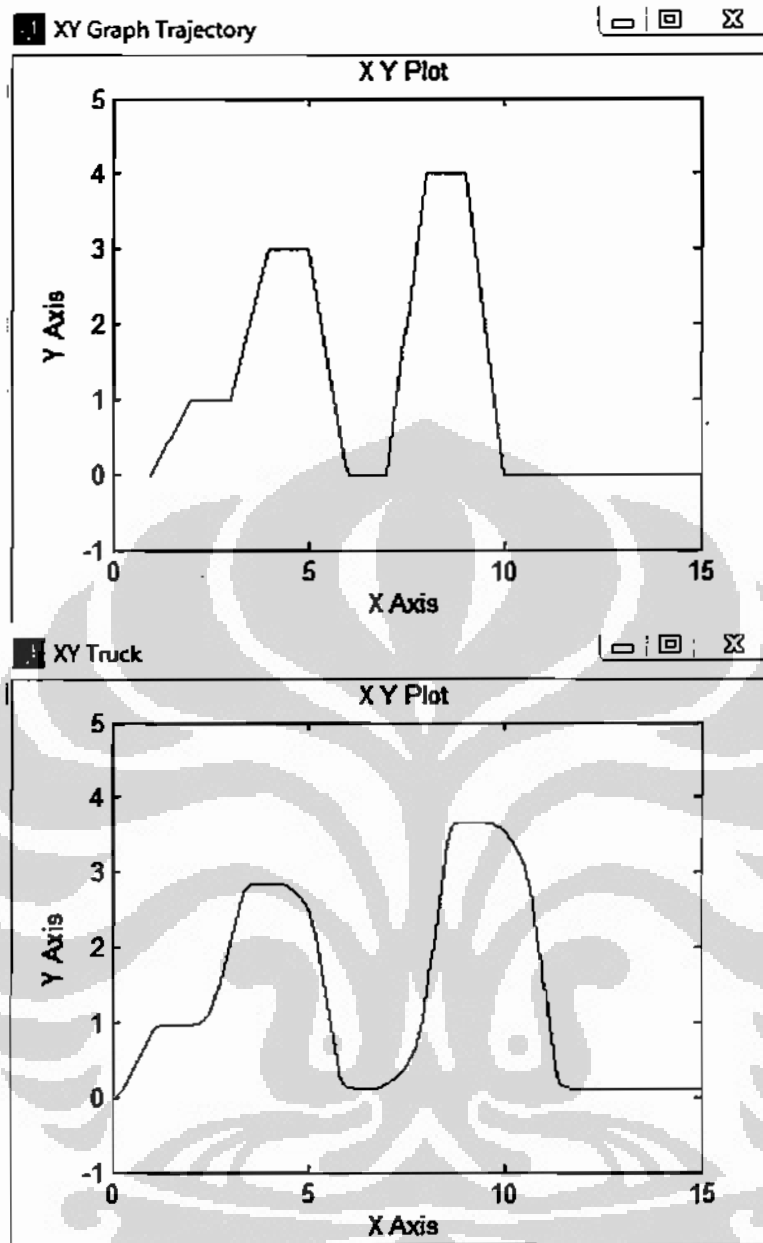


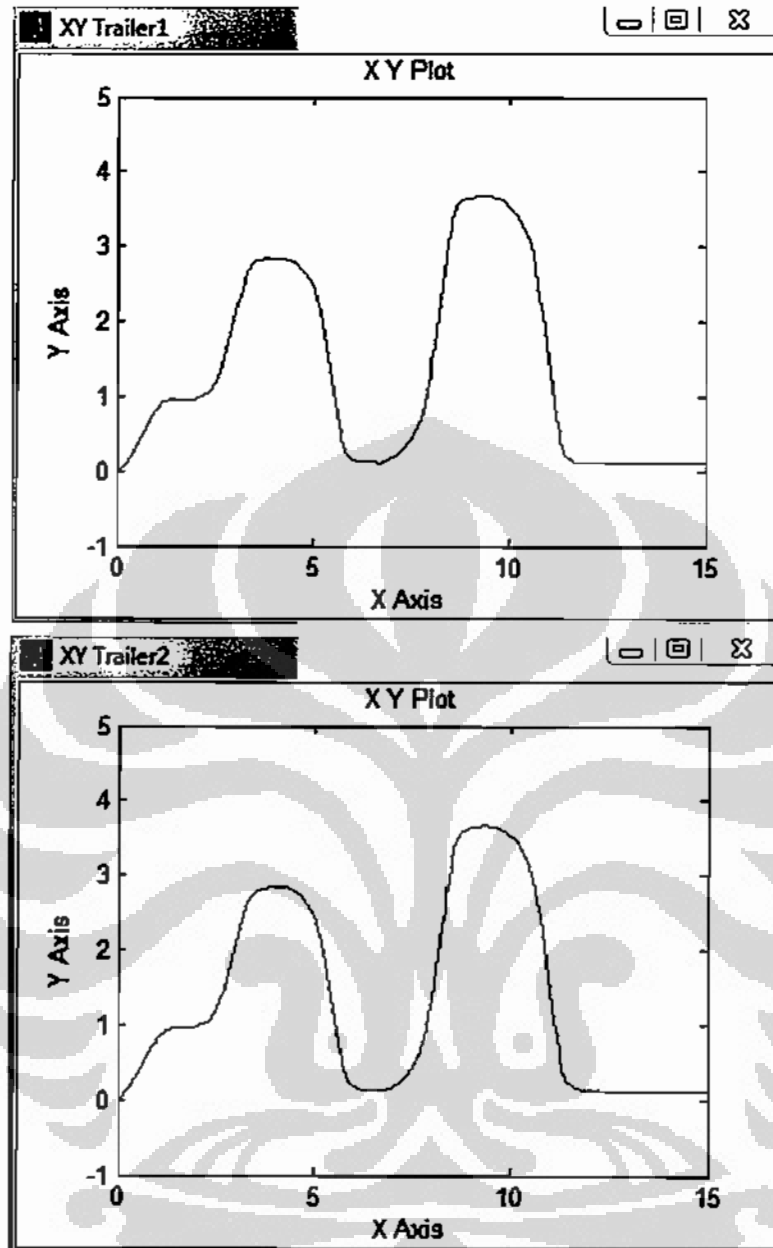
Gambar 2.7 Simulasi Mobile Robot-3 Trailers



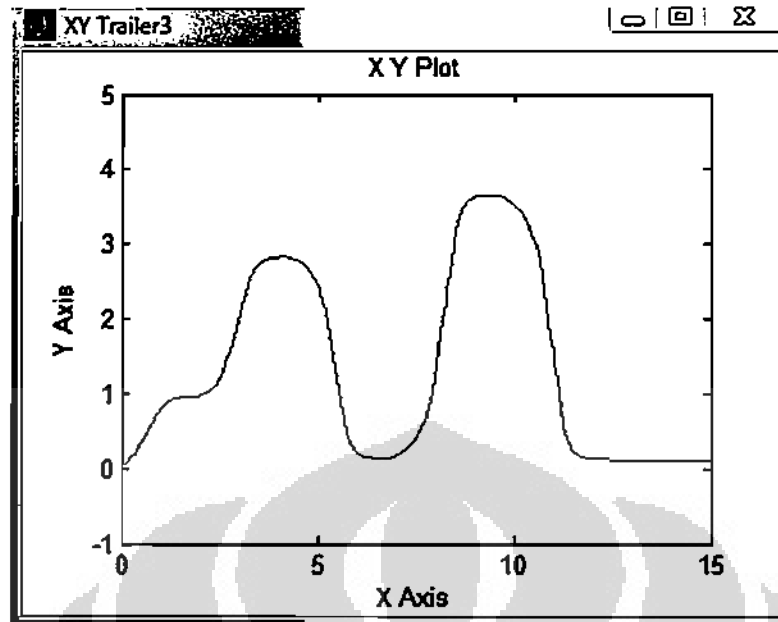
Gambar 2.8 Sudut Simulasi Mobile Robot-3 Trailers

Dari hasil simulasi sudut kemudi beresilasi ketika sudut putar *mobile robot* lebih besar dari  $45^\circ$ . Hal ini disebabkan oleh karena adanya pembatasan sudut kemudi *mobile robot*. Untuk lintasan dengan sudut yang lebih besar membuat konstanta waktu sistem menjadi lebih lambat.





Universitas Indonesia



Gambar 2.9 Posisi XY Simulasi Mobile Robot-3 Trailers

Gambar 2.9 menunjukkan grafik posisi *Mobile Robot-3 Trailers*.  
Kesalahan posisi yang terbesar terhadap *trajectory* dialami oleh *Trailer3*.



## BAB 3

### PERANCANGAN PERANGKAT KERAS DAN PERANGKAT LUNAK MOBILE ROBOT -3 TRAILERS

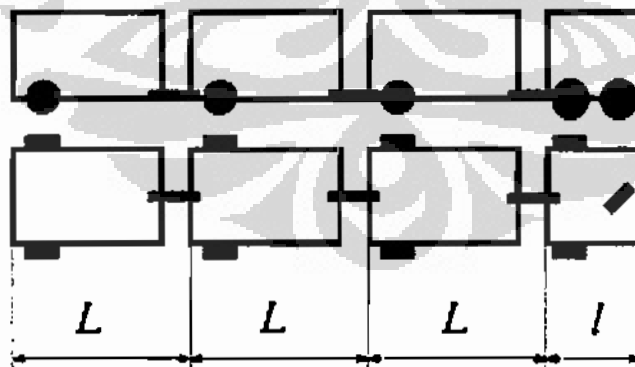
Perancangan perangkat keras terdiri dari pengukuran fisik *mobile robot-3 trailers*, pengenalan dan cara pengendalian perangkat *actuator* ( motor DC, motor servo) dan pengenalan dan cara pengendalian sensor (*rotary encoder, photoreflector dan potensiometer*).

Perancangan perangkat lunak terdiri dari simulasi dari program simulasi, pemograman mikrokontroler dan komunikasi antara program simulasi dan mikrokontroler.

#### 3.1 Perancangan Perangkat Keras

##### 3.1.1 Konstruksi Rangka

Gambar 3.1 menunjukkan konstruksi *Mobile Robot-3 Trailer*. Panjang *mobile robot* diukur dari depan *mobile robot* sampai ke poros gandingan *mobile robot* dengan trailer 1. Panjang *trailer* diukur dari poros gandingan depan dengan poros gandingan belakang.



Gambar 3.1 Mobile Robot-3 Trailer

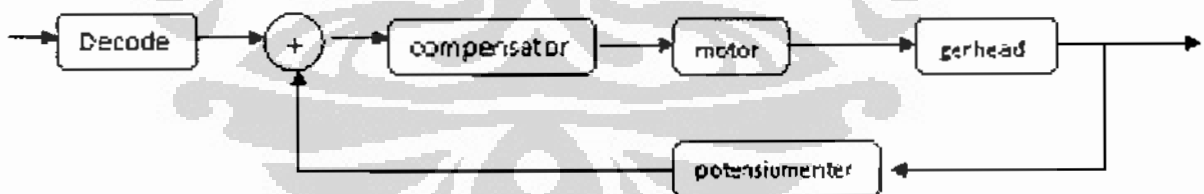
- Panjang poros roda depan ke titik sambungan dengan trailer ke-1 ( $l$ ) = 30 cm,

- Jarak antar sambungan ( $L$ ) atau jarak sambungan ke trailer terakhir = 15 cm,
- Diameter roda depan truk = 53 mm,
- Diameter roda belakang truk dan roda trailer lainnya = 56 mm,
- Jarak antar roda depan pada truk = 11 cm,
- Jarak antar roda belakang pada truk = 15 cm,
- Jarak antar roda pada masing-masing trailer = 10 cm.

Dalam perancangan model sistem, dimensi yang dipakai dalam perhitungan hanyalah panjang truk ( $D$ ), jarak sambungan ( $L$ ) dan diameter roda belakang truk (dipakai untuk menentukan kecepatan linear  $v_0$ ).

### 3.1.2 Aktuator (Motor Servo dan Motor DC)

Motor servo adalah sebuah motor dengan sistem umpan balik tertutup di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian gear, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor.



Gambar 3.2 Konstruksi Fisik Motor Servo

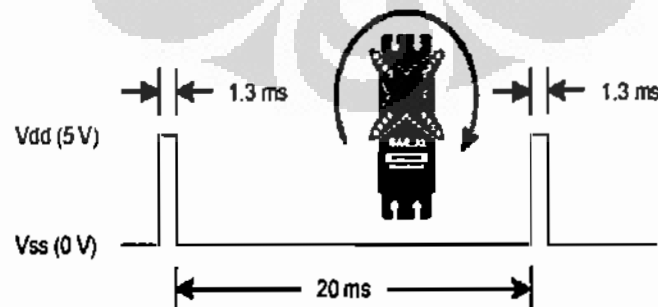
Karena motor DC servo merupakan alat untuk mengubah energi listrik menjadi energi mekanik, maka magnet permanent motor DC servolah yang mengubah energi listrik ke dalam energi mekanik melalui interaksi dari dua medan magnet. Salah satu medan dihasilkan oleh magnet permanent dan yang satunya dihasilkan oleh arus yang mengalir dalam

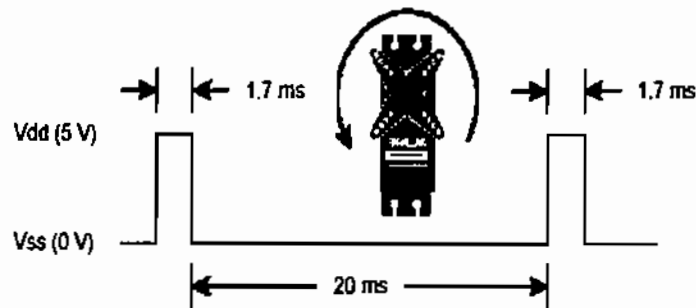
kumparan motor. Resultan dari dua medan magnet tersebut menghasilkan torsi yang membangkitkan putaran motor tersebut. Saat motor berputar, arus pada kumparan motor menghasilkan torsi yang nilainya konstan.

Secara umum terdapat 2 jenis motor servo. Yaitu motor servo standard dan motor servo Continuous. Servo motor tipe standar hanya mampu berputar 180 derajat. Motor servo standard sering dipakai pada sistem robotika misalnya untuk membuat "Robot Arm" ( Robot Lengan ). sedangkan Servo motor continuous dapat berputar sebesar 360 derajat. motor servo Continuous sering dipakai untuk Mobile Robot. Pada badan servo tertulis tipe servo yang bersangkutan.

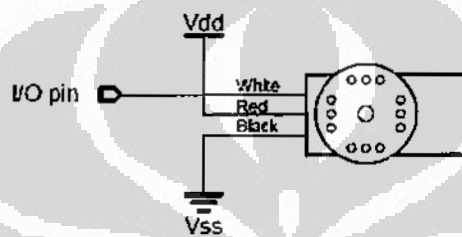
Motor servo merupakan sebuah motor de kecil yang diberi sistem gear dan potensiometer sehingga dia dapat menempatkan "horn" servo pada posisi yang dikehendaki. Karena motor ini menggunakan sistem close loop sehingga posisi "horn" yang dikehendaki bisa dipertahankan.

Pengendalian gerakan batang motor servo dapat dilakukan dengan menggunakan metode PWM. (Pulse Width Modulation). Teknik ini menggunakan sistem lebar pulsa untuk mengemudikan putaran motor. Sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Tampak pada gambar 3.3 dengan pulsa 1.5 mS pada periode selebar 2 mS maka sudut dari sumbu motor akan berada pada posisi tengah. Semakin lebar pulsa OFF maka akan semakin besar gerakan sumbu ke arah jarum jam dan semakin kecil pulsa OFF maka akan semakin besar gerakan sumbu ke arah yang berlawanan dengan jarum jam.





Gambar 3.3 Posisi Motor Servo



Gambar 3.4 Susunan kabel servo motor

Tabel 3.1 Spesifikasi teknis motor servo S3003 Futaba

Karakteristik	Besar
Sistem kontrol	kontrol pulsa dengan lebar 1520 $\mu\text{sec}$ untuk posisi Netral
Tegangan pulsa	3 – 5 volt puncak-puncak <i>square wave</i>
Tegangan operasi	4.8 – 6.0 volt
Range Temperatur	-20 – 60 derajat C
Kecepatan	0.23 sec/60 <sup>0</sup> (saat 4.8 volt) dan 0.19 scc/60 <sup>0</sup> (saat 6.0 volt)
Torsi	3.2 kg.cm (saat 4.8 volt) dan 4.1 kg.cm (saat 6.0 volt)
Sudut operasi	45 <sup>0</sup> putaran setiap sisi setiap lebar pulsa 400 $\mu\text{sec}$
Konsumsi arus	7.2 mA (4.8 volt tanpa beban) dan 8 mA (6.0 volt tanpa beban)

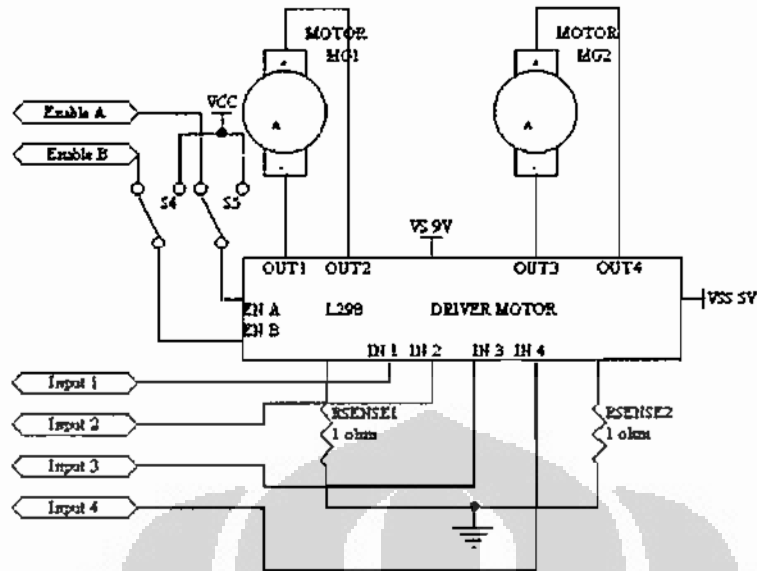
Dimensi	1.6" x 0.8"x 1.4" (41 x 20 x 36mm)
Berat	37.2 gram

Secara besaran waktu untuk membuat servo pada posisi center, diberikan pulsa 1.5ms. Untuk memutar servo ke kanan, diberikan pulsa  $\leq 1.3$ ms, dan pulsa  $\geq 1.7$ ms untuk berputar ke kiri dengan delay 20ms.

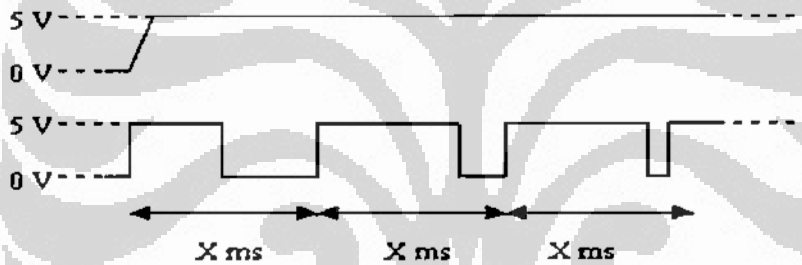
Sebuah motor DC seringkali digunakan sebagai divais penggerak dalam aplikasi robotika karena harganya relative murah, tetapi sayangnya untuk mengendalikan motor DC ini relatif rumit.

Ada beberapa macam driver motor DC yang biasa kita pakai seperti menggunakan relay yang diaktifkan dengan transistor sebagai saklar, namun yang demikian dianggap tidak efisien dan terlalu sulit dalam pengerjaan hardware-nya. Dengan berkembangnya dunia IC, sekarang sudah ada H Bridge yang dikemas dalam satu IC dimana memudahkan dalam pelaksanaan hardware dan kendalinya apalagi jika menggunakan mikrokontroler. IC yang familiar seperti ICL298 dan L298.

Modul yang menggunakan IC driver L298 yang memiliki kemampuan menggerakkan motor DC sampai arus 4A dan tegangan maksimum 46 VoltDC untuk satu kanalnya. Rangkaian driver motor DC dengan IC L298 diperlihatkan pada gambar 3.5. Pin *Enable A* dan *B* untuk mengendalikan jalan atau kecepatan motor, pin Input 1 sampai 4 untuk mengendalikan arah putaran. Pin *Enable* diberi VCC 5 Volt untuk kecepatan penuh dan PWM (*Pulse Width Modulation*) untuk kecepatan rotasi yang bervariasi tergantung dari level highnya. Ilustrasinya ditunjukkan pada gambar 3.6.



Gambar 3.5. Rangkaian Driver motor DC dengan L298



Gambar 3.6 ilustrasi Timing enable pada IC

Jika input 1 high(1) dan input 2 low(0) maka motor DC bergerak maju. Jika input 1 low(0) dan input 2 high(1) maka motor DC bergerak mundur. Sedangkan jika input 1 sama dengan input 2 maka motor DC berhenti. Jika input 3 high(1) dan input 4 low(0) maka motor DC bergerak maju. Jika input 3 low(0) dan input 4 high(1) maka motor DC bergerak mundur. Sedangkan jika input 3 sama dengan input 4 maka motor DC berhenti.

### 3.1.3. Sensor (Rotary encoder, Photoreflector dan Potensiometer)

Rotary encoder yang digunakan dalam perancangan ini dapat dilihat pada gambar dibawah ini:



Gambar 3.7 Rotary encoder

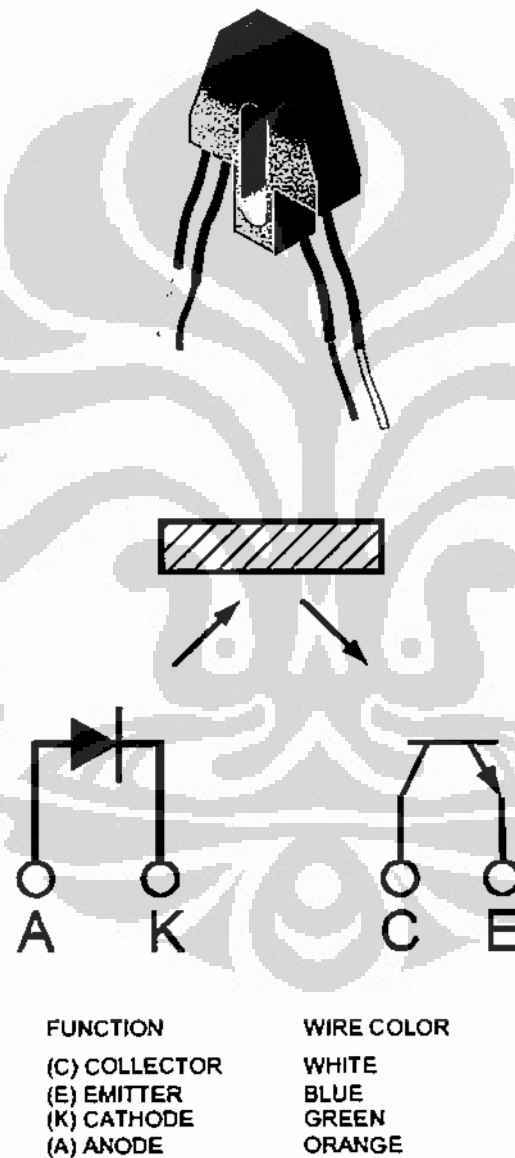
#### Spesifikasi:

- Piringan derajat dengan 36 lubang pada kelilingnya dengan sudut antara dua lubang yang berdampingan terhadap titik tengahnya adalah 10 derajat.
- Rangkaian sensor pembaca putaran yang menggunakan optocoupler tipe celah sebagai sensor pembaca perubahan posisi lubang piringan derajat.
- Sumber (VCC): 3,5 – 5,5V
- Logika output 0 (0 - 0.5 V) : Saat celah sensor terhalang
- Logika output 1 (3 - 5 V) : Saat celah sensor tanpa halangan
- Kecepatan baca sensor 1500Hz atau dengan 36 lubang berarti 2500 RPM

Pada perancangan ini digunakan sumber tegangan + 5 V DC sebagai sumber tegangan *rotary encoder*. Sinyal keluaran rotary encoder akan berlogika 0 saat celah sensor terhalang dan bernilai 1 saat celah sensor tanpa halangan. Piringan *rotary encoder* mempunyai 36 lubang sehingga beda antar lubang adalah 10 derajat. Untuk mengukur kecepatan *mobile robot*, panjang keliling roda *mobile robot* dikalibrasi dengan

jumlah pulsa *rotary encoder* dalam satu putaran. Sehingga diperoleh rumusan 1 pulsa rotary encoder bernilai berapa jauh

Kegunaan umum dari sensor *photoreflector* ini sebenarnya adalah sebagai pendeteksi kertas atau warna putih misal garis putih, tetapi tidak menutup kemungkinan digunakan untuk keperluan yang lain, misalnya sebagai saklar “sentuh” pada aplikasi ini.

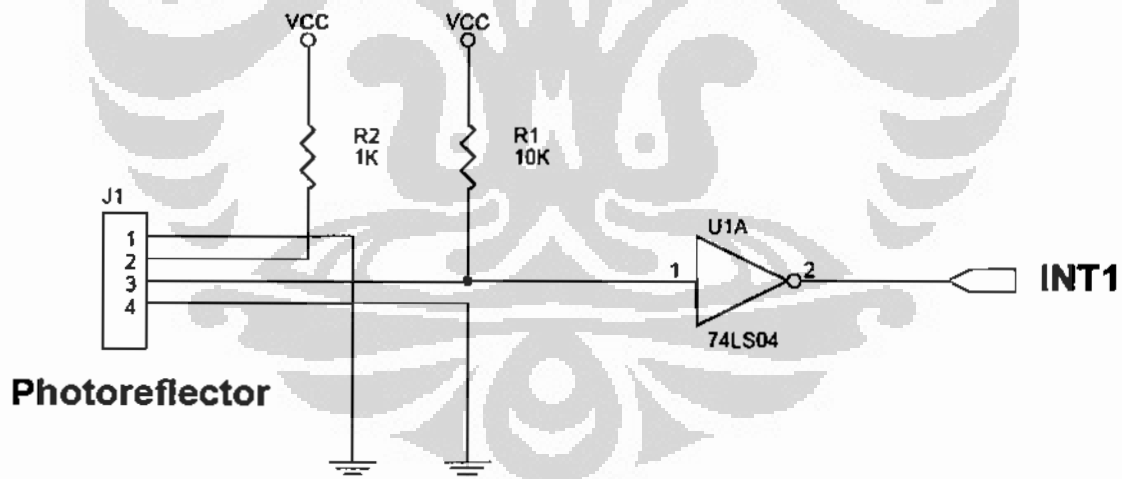


Gambar 3.8 *Photoreflector*



Sensor *photoreflector* yang digunakan pada prinsipnya menggunakan pantulan cahaya *Photoreflector* untuk mendeteksi dengan jarak tertentu adanya obyek berwarna hitam atau putih. Sensor *photoreflector* ini sangat sensitif sehingga juga dapat mendeteksi adanya obyek dengan warna lain selain hitam, semisal jari tangan. Didalam sensor *photoreflector* tersebut sudah terdapat build in photodiode, amplifier, Schmitt trigger dan transistor output.

Output dari sensor *photoreflector* sudah berbentuk digital yaitu hanya berlogika high atau low saja, sehingga dapat langsung diaplikasikan pada mikrokontroler. Jika mendeteksi obyek dengan warna selain hitam output sensor akan menghasilkan logika high, sebaliknya jika tidak mendeteksi obyek apa-apa maka output sensor akan berlogika low. Rangkaian internal dari sensor *photoreflector* adalah seperti pada gambar 3.8, Keluaran dari sensor *photoreflector* ini digunakan sebagai sinyal triger bagi mikrokontroler. Rangkaian interface sensor *photoreflector* dengan mikrokontroler adalah seperti pada gambar 3.9.

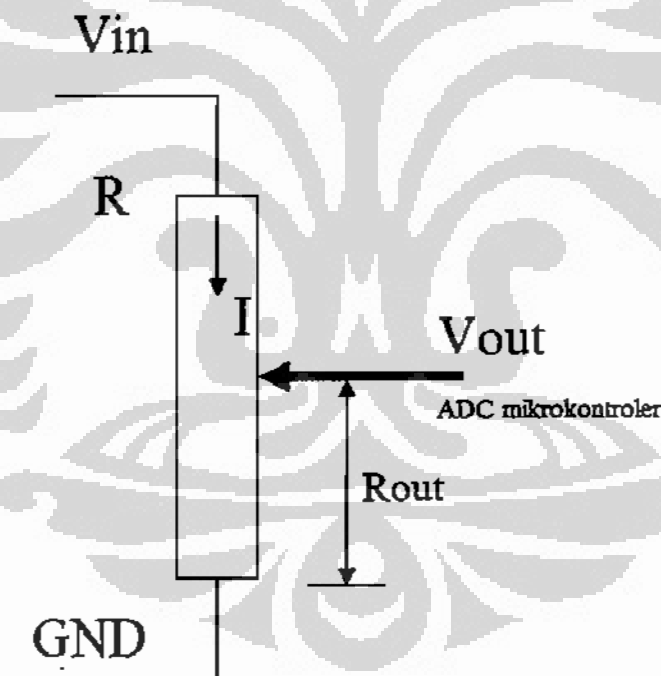


Gambar 3.9. Rangkaian Triger *Photoreflector* ke mikrokontroler

*Photoreflector* berjumlah dua buah diletakkan disisi kiri depan dan sisi kanan depan *mobile robot*. *photoreflector* disisi kiri dihubungkan ke pin interupsi 1 mikrokontroler ATMEGA 8535, sedangkan *photoreflector* disisi kanan dihubungkan ke pin interupsi 2 mikrokontroler ATMEGA

8535. jika *photoreflector* disisi kiri low (0) dan *photoreflector* disisi kanan high (1) maka *mobile robot* terbias ke kiri, oleh karena itu servo digerakkan ke arah kanan (CW). jika *photoreflector* disisi kiri high (1) dan *photoreflector* disisi kanan low (0) maka *mobile robot* terbias ke kanan, oleh karena itu servo digerakkan ke arah kiri (CCW).

Untuk mendeteksi sudut trailer digunakan potensiometer yang dihubungkan dengan port ADC (port A) mikrokontroler ATMEGA 8535 seperti terlihat pada gambar 3.10. pada perancangan ini digunakan 3 buah potensiometer 10 KOhm. Besarnya tegangan yang terbaca oleh ADC mikrokontroler ATMEGA 8535 akan dikonversikan dengan besaran sudut yang diperoleh dari hasil percobaan. Tegangan referensi yang digunakan adalah 5 Volt DC sehingga setiap kenaikan 0.5 mVolt mewakili 1 Ohm potensiometer.



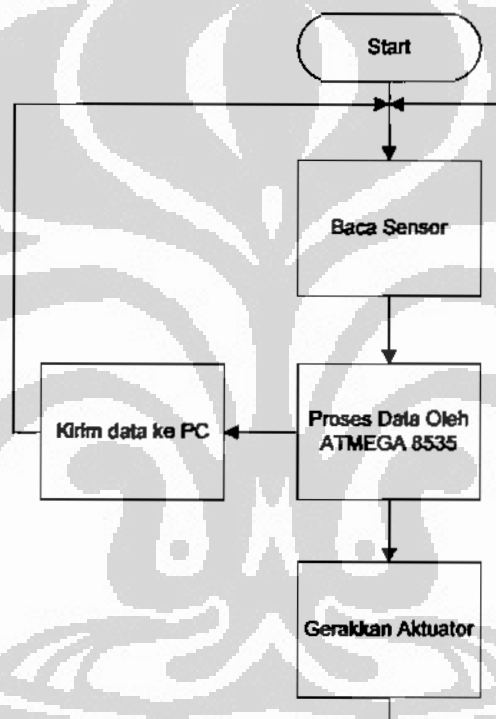
Gambar 3.10. Rangkaian Kendali Sudut dengan Potensiometer

### 3.2 Perangkat Lunak

Perancangan perangkat lunak terdiri dari perancangan model menggunakan program simulasi, perancangan program mikrokontroler dengan bahasa C dan program koneksi antara program simulasi dan mikrokontroler.

#### 3.2.1 Program mikrokontroler ATMEGA 8535

Secara garis besar perancangan perangkat lunak digambarkan sebagai berikut:



Gambar 3.11 Flowehart Perancangan Perangkat Lunak

Perancangan perangkat lunak dari sisi mikrokontroler menggunakan bahasa C. program terdiri dari: program pengendalian motor DC, pengendalian motor Servo, Pembacaan sensor Rotary encoder, Pembacaan Sensor *Photoreflector*, Pembacaan ADC untuk mendeteksi Sudut.

Program motor DC adalah untuk mengendalikan kecepatan motor DC. Dari gambar 3.5 kecepatan motor DC dikendalikan dari pin enable I298. Pin enable ini dihubungkan pin PB3 mikrokontroler ATMEGA 8535. Pin PB3 mikrokontroler ATMEGA 8535 akan membangkitkan sinyal PWM dari timer 0. Speed dari motor DC akan dibaca oleh rotary encoder yang menghasilkan sinyal pulsa melalui interrupt 0 pin PD2. Dalam perancangan ini rotary encoder yang digunakan 36 pulsa per putaran. Nilai pulsa ini akan dikonversikan menjadi kecepatan dengan perhitungan sesuai dengan keliling *roda mobile robot*. Kecepatan dari rotary encoder ini merupakan umpan balik yang akan dibandingkan dengan kecepatan mobile robot yang diinginkan.



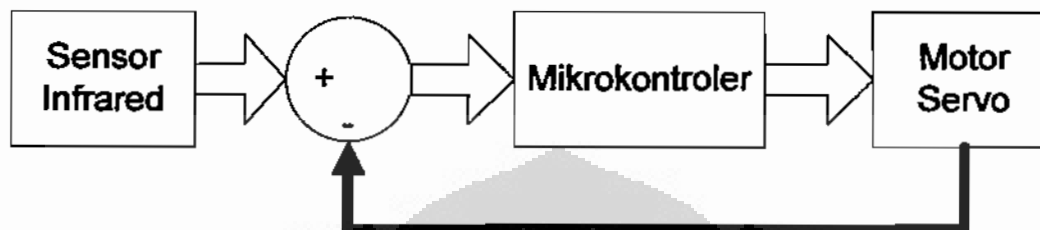
Gambar 3.12 Pengendalian Motor DC

Setiap pulsa rotary encoder akan menginterupsi mikrokontroler ATMEGA 8535 yang akan menaikkan cacahan. Nilai variable pulsa akan bertambah jika arah rotary encoder maju dan akan berkurang jika arah encoder mundur. Nilai cacahan akan dikonversi ke kecepatan dengan perubahan waktu setiap 102.4 ms.

Nilai OCR0 yang berisi nilai PWM yang dihasilkan mikrokontroler ATMEGA 8535 akan bertambah jika kecepatan *mobile robot* lebih kecil dari kecepatan yang diinginkan. Sebaliknya nilai OCR0 akan berkurang jika kecepatan *mobile robot* lebih besar dari kecepatan yang diinginkan. Kondisi ini akan berulang sampai kecepatan *mobile robot* sama dengan kecepatan yang diinginkan.

Gerak motor servo mempengaruhi arah gerak mobile robot. Pulsa PWM yang menggerakkan motor servo diberikan oleh timer 1 (OCR1A)

mikrokontroler ATMEGA 8535. Titik tengah motor servo dari datasheet motor servo bernilai 1520  $\mu\text{sec}$ . Jika bergerak ke kanan (CW) nilai yang diberikan ke OCR1A lebih kecil dari 1520  $\mu\text{sec}$ . Jika bergerak ke kiri (CCW) nilai yang diberikan ke OCR1A lebih besar dari 1520  $\mu\text{sec}$



Gambar 3.13 Pengendalian Motor Servo

Pergerakan motor servo ke kiri atau ke kanan akan dibandingkan dengan keluaran sensor *photoreflector*. jika *photoreflector* disisi kiri low (0) dan *photoreflector* disisi kanan high (1) maka motor servo digerakkan ke arah kanan (CW). jika *photoreflector* disisi kiri high (1) dan *photoreflector* disisi kanan low (0) motor servo digerakkan ke arah kiri (CCW).

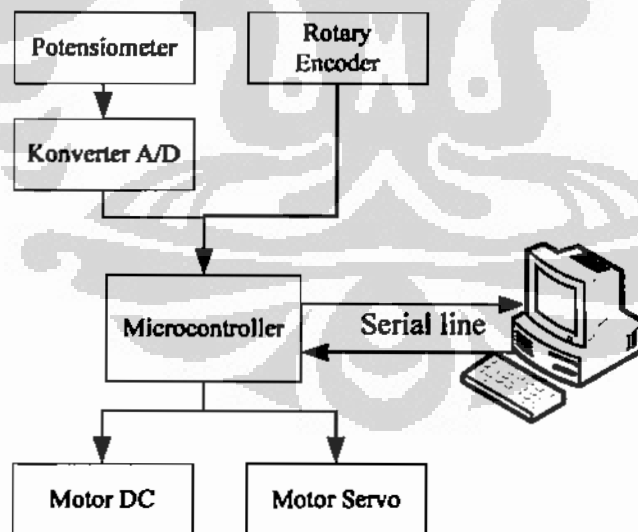
Pergerakan motor servo ke kiri atau ke kanan akan dibandingkan dengan keluaran sensor *photoreflector*. jika *photoreflector* disisi kiri low (0) dan *photoreflector* disisi kanan high (1) maka motor servo digerakkan ke arah kanan (CW). jika *photoreflector* disisi kiri high (1) dan *photoreflector* disisi kanan low (0) motor servo digerakkan ke arah kiri (CCW).

Satu derajat motor servo bernilai 8.89 ms. Sudut kemudi ditentukan sesuai dengan sudut yang diinginkan. Nilai PWM untuk menggerakkan motor servo dibangkitkan oleh timer 1 (OCR1A) mikrokontroler ATMEGA 8535. Jika tidak terjadi interupsi timer 1 dan timer 2, maka posisi motor berada ditengah. Ini berarti *mobile robot* berada di jalur (*trajectory*) yang telah ditentukan.

Interrupt 1 akan berulang-ulang selama *mobile robot* keluar dari jalur kearah kiri, sehingga motor servo digerakkan kearah kanan (CW). Interrupt 2 akan berulang-ulang selama *mobile robot* keluar dari jalur kearah kanan, sehingga motor servo digerakkan kearah kiri (CCW).

Satu derajat motor servo bernilai 8.89 ms. Sudut kemudi ditentukan sesuai dengan sudut yang diinginkan. Nilai PWM untuk menggerakkan motor servo dibangkitkan oleh timer 1 (OCR1A) mikrokontroler ATMEGA 8535. Jika tidak terjadi interupsi timer 1 dan timer 2, maka posisi motor berada ditengah. Ini berarti *mobile robot* berada di jalur (*trajectory*) yang telah ditentukan.

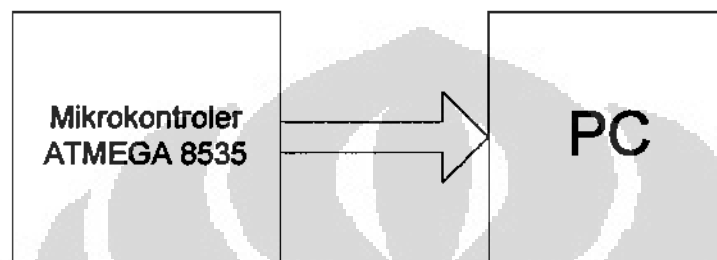
Pada simulasi Mobile robot-3 Trailers bergerak mengikuti *trajectory* yang diterima dari sensor *photoreflector*. Kecepatan mobile robot dari keadaan diam, digerakkan oleh motor DC sampai kecepatan yang telah ditentukan. Kecepatan mobile robot dibaca oleh *rotary encoder*. Sudut kemudi mobile robot digerakkan oleh motor servo. Data sudut kemudi, kecepatan mobile robot dan sudut trailer dikirim ke PC menggunakan komunikasi serial. Flowchart integrasi Mobile Robot-3 Trailers seperti gambar 3.14.



Gambar 3.14. Flowchart Mobile Robot-3 Trailers

### 3.2.2. Komunikasi PC dengan Mikrokontroler

Komunikasi PC dengan Mikrokontroler menggunakan komunikasi serial. mikrokontroler ATMEGA 8535 memiliki kemampuan komunikasi serial melalu PORTD.0 dan PPRTD.1. Data yang dikirimkan dari mikrokontroler ATMEGA 8535 ke PC adalah sudut kemudi *mobile robot* dan kecepatan *mobile robot*.



Gambar 3.15 Komunikasi serial ATMEGA 8535 dengan PC

Data sudut kemudi yang disimpan dalam variabel data dikirim dengan komunikasi serial *Asynchronous, 8 Data, 1 Stop, No Parity* dan *Baud Rate 9600*. Dikirim Hal yang sama data kecepatan *mobile robot* yang disimpan dalam variable data1 dikirim dengan komunikasi serial *Asynchronous, 8 Data, 1 Stop, No Parity* dan *Baud Rate 9600*. Dari sisi PC harus komunikasi serialnya harus *disetting* sesuai dengan *setting* dari sisi Mikrokontroler ATMEGA 8535.

## BAB 4

### SIMULASI DAN UJI COBA KOMPONEN

#### 4.1. Simulasi dan Pengujian Motor Servo untuk Kemudi Truck

Pengujian motor servo menggunakan tegangan 5 VDC. Dari datasheet motor servo S3003 Futaba untuk menggerakkan motor servo ke posisi tengah dibutuhkan lebar pulsa sebesar 1520  $\mu\text{sec}$ . Motor servo bergerak  $45^\circ$  putaran setiap sisi setiap lebar pulsa 400  $\mu\text{sec}$ , sehingga setiap derajat putarannya adalah:

$$1^\circ = \frac{400 \mu\text{sec}}{45^\circ} = 8.889 \mu\text{sec}$$

Dalam pengujian, sudut kemudi ditentukan sesuai dengan sudut yang diinginkan. Nilai PWM untuk menggerakkan motor servo dibangkitkan oleh timer 1 (OCR1A) mikrokontroler ATMEGA 8535. Sudut kemudi yang telah ditentukan dikonversikan ke satuan waktu dengan rumusan sebagai berikut:

$$\begin{aligned} t_{\text{pulsa}} &= 1520 + \text{rot\_1deg} * \text{sdt\_kemudi}; \\ \text{OCR1A} &= t_{\text{pls}}/2; \end{aligned}$$

Dimana  $\text{rot\_1deg}$  adalah 8.889 dan  $\text{sdt\_kemudi}$  adalah sudut kemudi mobile robot dalam derajat.

Sebagai contoh untuk bergerak lurus berarti sudut kemudi bernilai 0, maka nilai  $\text{OCR1A} = 760$ .

Program pengendalian motor servo dapat dilihat dibawah ini:

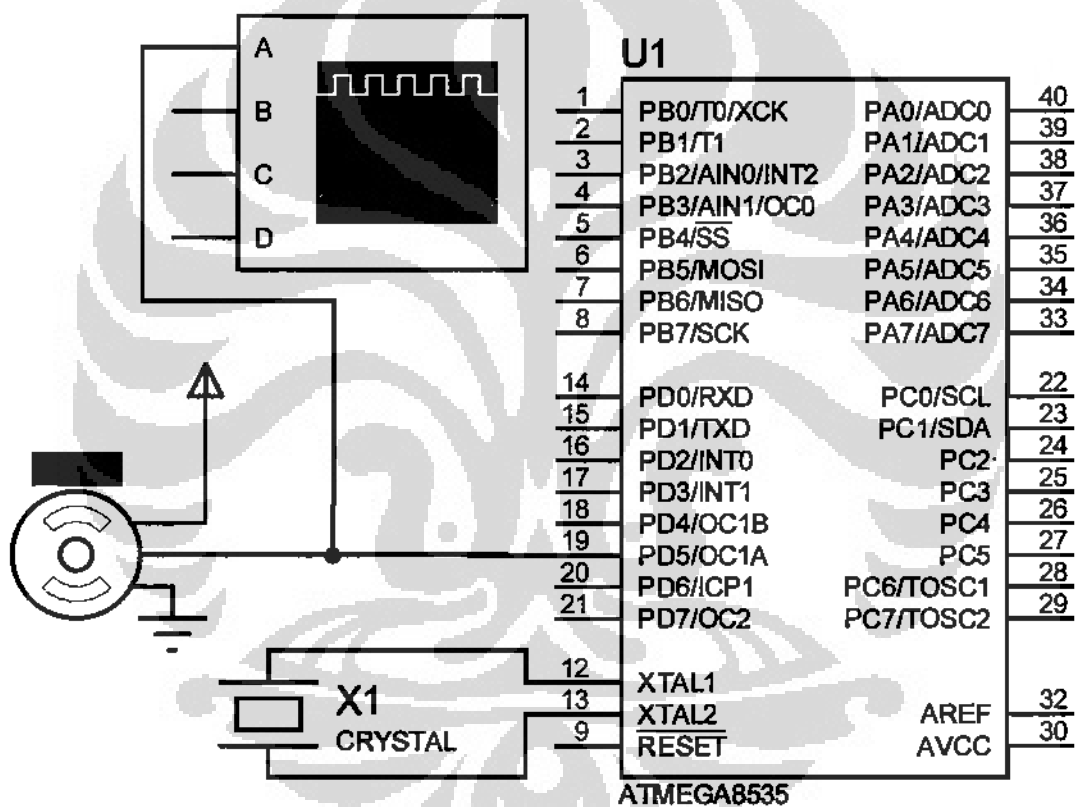
```
//putaran ke kanan (CW)
}
t_pls = 1520 + rot_1deg * sdt_kemudi;
OCR1A=t_pls/2;
}
//putaran ke kiri (CCW)
```



```

}
    t_pls = 1520 + rot_1deg * sdt_kemudi;
    OCR1A=t_pls/2;
}
//putaran ke tengah (netral)
{
    {OCR1A=760;}           //posisi tengah servo motor
(1520 us)
}

```



Gambar 4.1. Rangkaian pengujian Motor Servo

Sinyal keluaran PWM OCR1A dihubungkan ke sinyal masukan motor servo dan dihubungkan ke osiloskop untuk membuktikan sinyal PWMnya.

Hasil percobaan dapat dilihat pada table berikut:

Tabel 4.1 Percobaan Motor Servo

no	Sudut kemudi (°)	Nilai OCR1A	Kondisi <i>real</i> (°)	Simulasi (°)
1	10	804	11	13,2
2	20	848	22	23.8
3	30	893	32	34.6
4	-10	715	-9	-8.16
5	-20	671	-19	-18.7

Pengukuran kondisi *real* pada table diatas menggunakan busur derajat dan hasil simulasi diperoleh dari program simulasi computer.

Pada kondisi *real*, kesalahan dari nilai sudut kemudi yang diinginkan terjadi akibat adanya beban yang dialami motor servo dan pembulatan nilai OCR1A yang dilakukan program simulasi. Sedangkan dalam simulasi kesalahan terjadi pembulatan nilai OCR1A yang dilakukan program simulasi.

#### 4.2. Simulasi Motor DC dengan Feedback Encoder

Kecepatan motor DC dihitung dari sensor *rotary encoder*. *rotary encoder* yang digunakan simulasi ini adalah piringan derajat dengan 36 lubang pada kelilingnya dengan sudut antara dua lubang yang berdampingan terhadap titik tengahnya adalah 10 derajat.

Program simulasi motor DC dengan *feedback encoder* dapat dilihat dibawah ini:

```
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
```

```

{
if (a>0)
    pulsa++;
else
    pulsa--;
}

// Timer 2 overflow interrupt service routine intrupt setiap 51,2 ms
// pencuplikan setiap 102.4ms
interrupt [TIM2_OVF] void timer2_ovf_isr(void)
{
waktu++;
if (waktu%2==0)
    {speed = 0.004888889*(float)(pulsa-pulsaold)/0.1024;
    //keliling roda/36 pulsa=0.00488
    pulsaold=pulsa;
if (speed<vref)
    {OCR0++;}
if (speed>vref)
    { OCR0--;}
}
}

```

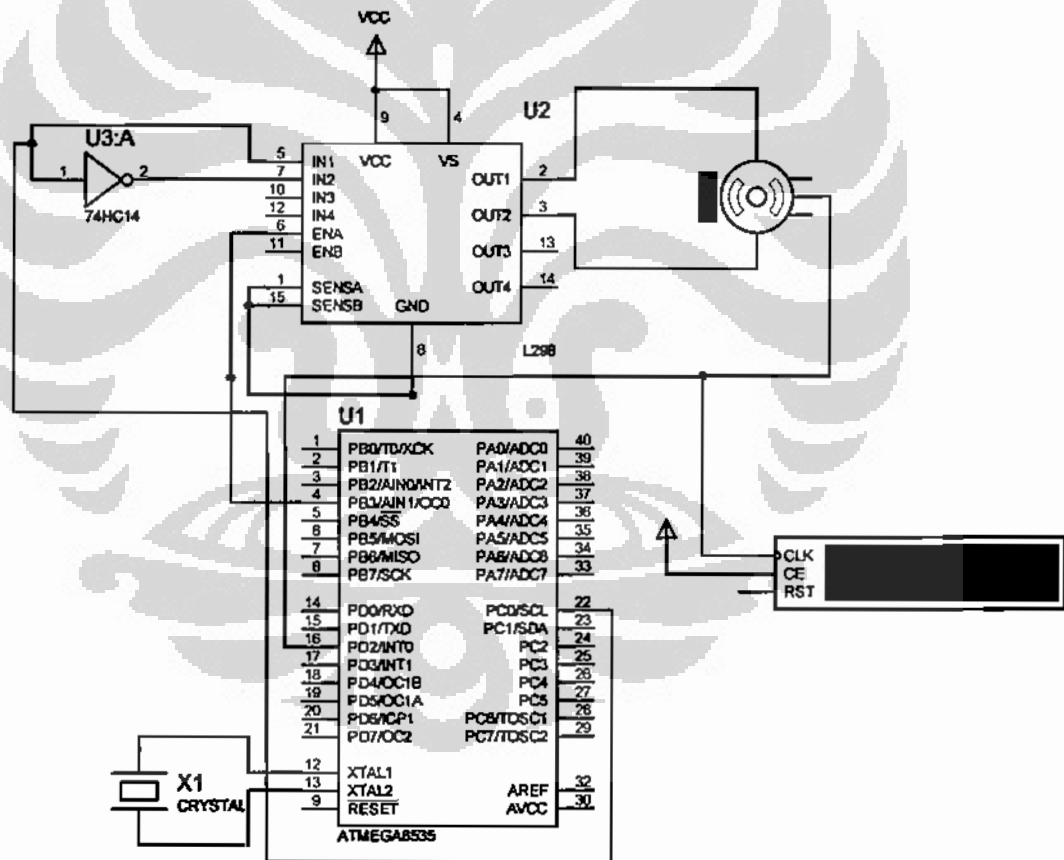
Pulsa rotari encoder dihitung melalui interupsi 0 mikrokontroler ATMEGA 8535. Pulsa bertambah atau berkurang ditentukan oleh arah pergerakan *mobile robot*. pulsa yang terbaca akan digunakan untuk menghitung kecepatan motor DC. Kecepatan motor DC yang terbaca akan dibandingkan dengan kecepatan referensi. Jika kecepatan motor DC lebih kecil dari kecepatan referensi maka nilai OCR0 akan bertambah. Sedangkan jika kecepatan motor DC lebih besar dari kecepatan referensi maka nilai OCR0 akan berkurang.

Tabel 4.2. Simulasi Motor DC dengan Feedback Encoder

Vref (m/s)	OCR0(maks 255)	Duty cycle(%)	Vsimulasi(m/s)
0,05	90	35,3	0,047
0,1	107	42	0,094
0,15	133	52,1	0,141
0,2	160	62,7	0,188
0,25	195	76,4	0,235

Dari table diatas dapat dilihat bahwa nilai kecepatan simulasi lebih kecil dari nilai kecepatan referensi yang diberikan. Ini disebabkan oleh pembulatan nilai kecepatan simulasi yang dilakukan program sesuai dengan tipe variable yang digunakan.

Gambar rangkaian pengujian Motor DC dengan feedback encoder dapat dilihat pada gambar dibawah ini:



Gambar 4.2. Rangkaian Pengujian Motor DC dengan feedback Encoder

Pada simulasi disertakan juga *instrument* pembaca pulsa untuk dapat mengetahui langsung nilai pulsa dari *rotary encoder*.

#### 4.3. Simulasi dan Pengujian Sensor Photoreflector

Pada simulasi ini sensor *Photoreflector* diwakilkan dengan *switch*. Kondisi *switch* aktif pada saat terang. Sensor photoreflector berjumlah dua buah diletakkan disisi kiri depan dan sisi kanan depan *mobile robot*. Sensor *photoreflector* disisi kiri dihubungkan ke pin interupsi 1 mikrokontroler ATMEGA 8535, sedangkan sensor *photoreflector* disisi kanan dihubungkan ke pin interupsi 2 mikrokontroler ATMEGA 8535. jika sensor *photoreflector* disisi kiri (S1) low (0) dan sensor *photoreflector* disisi kanan (S2) high (1) maka *mobile robot* terbias ke kiri, oleh karena itu servo digerakkan ke arah kanan (CW). jika sensor *photoreflector* disisi kiri (S1) high (1) dan sensor *photoreflector* disisi kanan (S2) low (0) maka *mobile robot* terbias ke kanan, oleh karena itu servo digerakkan ke arah kiri (CCW). Program pengendalian sensor *photoreflector* dapat dilihat dalam program berikut.

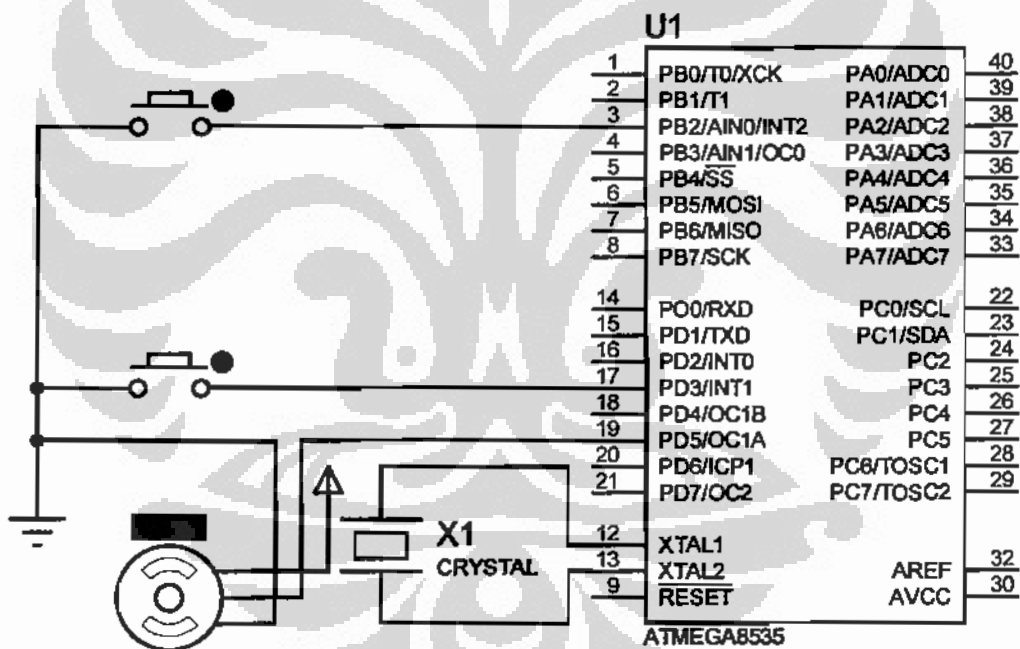
```
#define rot_1deg 400/45;
// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
  cw();
}
// External Interrupt 2 service routine
interrupt [EXT_INT2] void ext_int2_isr(void)
{
  ccw();
}
void netral()
{OCR1A=760;} //1520 us
```

```

void ccw()          //putaran ke kiri
{
    t_pls = 1520 + rot_1deg * sdt_kemudi;
    OCR1A=t_pls/2;
}

void cw()           //putaran ke kanan
{
    t_pls = 1520 + rot_1deg * sdt_kemudi;
    OCR1A=t_pls/2;
}

```



Gambar 4.3 Rangkaian Pengujian Photoreflector

Pada simulasi ini setiap interupsi yang dilakukan photoreflector akan menggerakkan sudut servo sebesar 10 derajat. Besar sudut mobile

robot terhadap sumbu X tergantung dari berapa lama *photoreflector* menginterupsi ATMEGA 8535.

Tabel 4.3 Pengujian Photoreflector

Kondisi Switch		Sudut Motor servo (Real)	Sudut Motor servo (Simulasi)
S1	S2		
0	0	-	-
0	1	-9	-8.16
1	0	11	13,2
1	1	0	0

Dari hasil *datasheet*, sensor *photoreflector* mempunyai respon yang baik jika jarak antara sensor *photoreflector* dan bidang pantul lebih kecil dari 3,8 mm (0,15 inches).

#### 4.4. Pengujian Sensor Sudut Trailer dengan potensiometer

Potensiometer digunakan untuk mengukur sudut trailer. Data yang terbaca oleh ADC mikrokontroler ATMEGA 8535 akan dikonversikan menjadi besaran sudut. Teknik pengkonversian ADC menggunakan data 8 Bit. Dengan menggunakan potensiometer 10 KOhm maka setiap kenaikan cacah ADC setara dengan kenaikan 39 Ohm.

Program pembacaan data ADC dapat dilihat dibawah ini:

```
#define ADC_VREF_TYPE 0x20

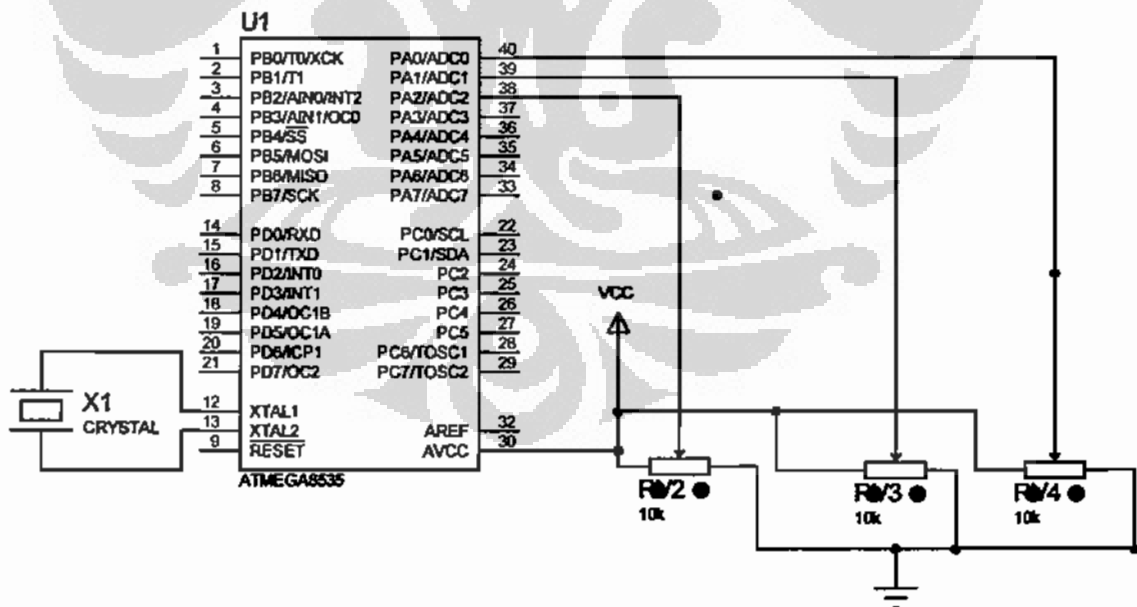
// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
  ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
```

```

// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCH;
}
while (1)
{
x=read_adc(0);      // Baca sudut trailer 1
y=read_adc(1);     // Baca sudut trailer 2
z=read_adc(2);     // Baca sudut trailer 3
};

```

Gambar rangkaian pembacaan sensor ADC dapat dilihat pada gambar dibawah ini.



Gambar 4.4. Rangkaian pengujian Potensiometer

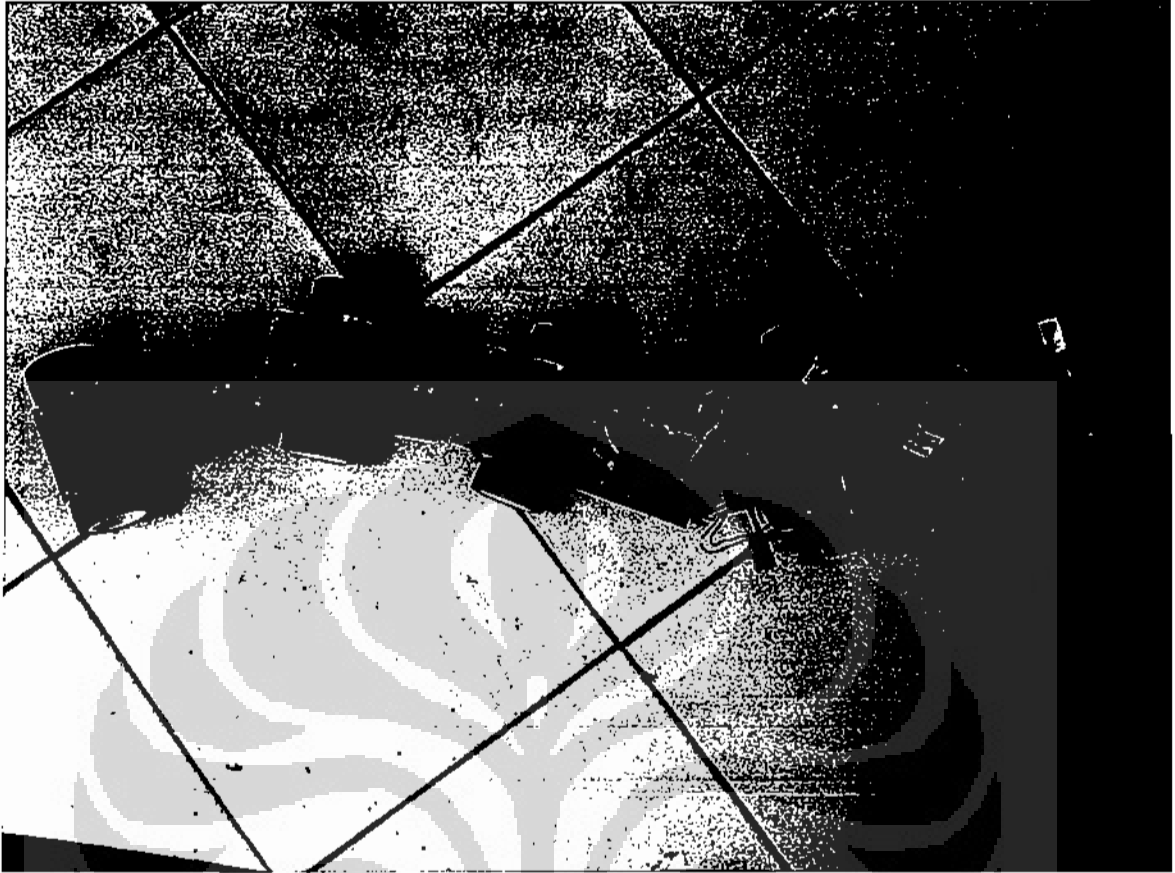


Potensiometer yang digunakan pada percobaan ini adalah 10 KOhm dengan 1 putaran penuh sebesar 270 derajat. Posisi nol derajat trailer adalah posisi tengah putaran potensiometer (5KOhm). Setiap perubahan 1 Ohm potensiometer akan terbaca oleh ADC ATMEGA 8535 0.5 mVolt.

Tabel 4.4 Pengujian Sensor Sudut Trailer dengan potensiometer

Sudut	Nilai ADC	Perhitungan Sudut
0	127	1
10	137	11
20	148	20
30	160	31
-10	116	-9
-20	106	-19

Pengujian diatas menggunakan busur derajat dalam perhitungan sudut. Kesalahan perhitungan dari nilai sudut yang dikehendaki disebabkan oleh ketelitian pengukuran dan tingkat ketelitian konversi ADC ATMEGA 8535 yang menggunakan teknik pengkonversian ADC menggunakan data 8 Bit.



Gambar 4.5. Model Mobile Robot-3 Trailers

## BAB 5

### KESIMPULAN

Dari keseluruhan pembahasan dalam skripsi ini dapat disimpulkan beberapa hal, yaitu :

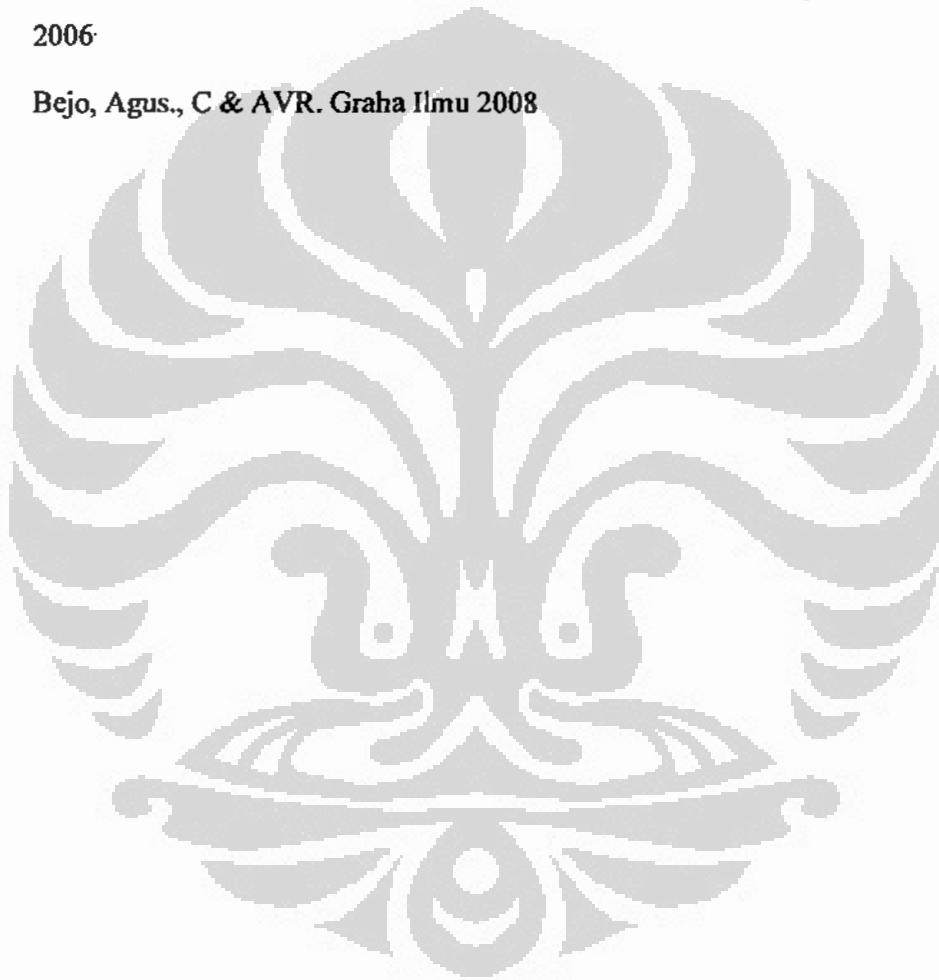
1. Sistem nonlinear truk-trailer dapat diturunkan menjadi pendekatan orde 1.
2. Hasil simulasi dengan program simulasi menunjukkan kesalahan yang terbesar terhadap *trajectory* dialami trailer<sup>3</sup>.
3. Fasilitas komunikasi serial ATMEGA 8535 dapat digunakan untuk *interface* antara sensor dan masukkan model sistem nonlinear truk-trailer .
4. Motor servo S3003 Futaba memiliki keakuratan yang tinggi.
5. Motor DC yang digunakan pada percobaan ini memiliki torsi yang besar, namun kecepatannya rendah.
6. ATMEGA 8535 memiliki *fitur-fitur* yang lain dan dalam perancangan ini masih banyak I/O yang belum digunakan yang dapat digunakan untuk pengembangan lebih lanjut.
7. Jarak pembacaan sensor *photoreflector* sangat kecil, sehingga tidak dapat digunakan untuk lintasan yang bergelombang.

**DAFTAR REFERENSI**

1. J. Laumond., *Robot Motion Planning and Control*, Springer, 1998.
2. Ogata, Katsuhito., *Modern Control Engineering*, Prentice Hall.2002
3. Nise, Norman S., *Control System Engineering*. John Wiley & Sons. 2003.
4. G.Oriolo., *Mobile Robotics: Control Problems*.2002
5. Pitowarno, Endra., *Robotika*. ANDI Yokyakarta.2006.
6. Palmer, Ed., *Introduction to Control System*. School of Electrical & Elektronik System Engineering.
7. Wahab, Wahidin., *Diktat Kuliah Dasar Sistem Kendali*. Elektro FTUI, 2001
8. Ogata, K. *Solving Control Engineering Problems with MATLAB*, Prentice-Hall.1994
9. M. Thoma. M. Morari. *Robot Motion and Control*. Springer. 2006
10. Budiharto, Widodo., *Membuat Robot Cerdas*. Elex Media Komputindo 2006
11. Bcjo, Agus., *C & AVR*. Graha Ilmu 2008
12. Heryanto, Ary., *Pemograman Bahasa C*. ANDI Yokyakarta 2008

**DAFTAR ACUAN**

1. G.Oriolo., *Mobile Robotics: Control Problems*.2002
2. J. Laumond., *Robot Motion Planning and Control*, Springer, 1998.
3. M. Thoma. M. Morari. *Robot Motion and Control*. Springer. 2006
4. Budiharto, Widodo., *Membuat Robot Cerdas*. Elex Media Komputindo 2006
5. Bejo, Agus., *C & AVR*. Graha Ilmu 2008



## LAMPIRAN

### Program Simulasi Truck-Trailer

```

function [sys, x0] = animasic3t(t, x, u, flag, action)
global AnimTbuFigH AnimTbuFigTitle
global TbuTruck TbuTrail

if ~isempty(flag) & flag == 3, % return the output value
    sys = 0;
    x0=[];
elseif ~isempty(flag) & flag == 2,
    if any(get(0, 'children') == AnimTbuFigH),
        if strcmp(get(AnimTbuFigH, 'Name'), AnimTbuFigTitle),
            xPos = u(1); yPos = u(2); phi = u(3);
            x1Pos = u(4); y1Pos = u(5); phi1 = u(6);
            x2Pos = u(7); y2Pos = u(8); phi2 = u(9);
            x3Pos = u(10); y3Pos = u(11); phi3 = u(12);
            %if length(u)==4, theta = u(4); else theta = 0; end
            curr_pos = xPos + j*yPos;
            curr_pos1 = x1Pos + j*y1Pos;
            curr_pos2 = x2Pos + j*y2Pos;
            curr_pos3 = x3Pos + j*y3Pos;
            tmp = get(AnimTbuFigH, 'UserData');
            truckH=tmp(1,:);
            trail1H=tmp(2,:);
            trail2H=tmp(3,:);
            trail3H=tmp(4,:);

            truck_l = 2*abs(real(TbuTruck(1)));
            truck_w = 2*abs(imag(TbuTruck(1)));
            trail_l = 2*(abs(real(TbuTrail(1))));
            trail_w = 2*(abs(imag(TbuTrail(1))));

            new_truck = (TbuTruck + truck_l/2)*exp(j*phi) +
curr_pos;
            new_trail1 = (TbuTrail + trail_l/2)*exp(j*phi1) +
curr_pos1;
            new_trail2 = (TbuTrail + trail_l/2)*exp(j*phi2) +
curr_pos2;
            new_trail3 = (TbuTrail + trail_l/2)*exp(j*phi3) +
curr_pos3;

            set(truckH, 'XData', real(new_truck), 'YData',
imag(new_truck));
            set(trail1H, 'XData', real(new_trail1), 'YData',
imag(new_trail1));
            set(trail2H, 'XData', real(new_trail2), 'YData',
imag(new_trail2));

```

```

        set(trail3H, 'XData', real(new_trail3), 'YData',
imag(new_trail3));
    end
end
% ===== return nothing
sys = [];
x0=[];
elseif ~isempty(flag) & flag == 9, % When simulation stops ...
% ===== change labels of standard UI controls
if any(get(0, 'children') == AnimTbuFigH),
    if strcmp(get(AnimTbuFigH, 'Name'), AnimTbuFigTitle),
        end
    end
elseif ~isempty(flag) & flag == 0,
% ===== find animation block & figure
[winName] = bdroot(gcs);
AnimTbuFigTitle = [winName, ': Animasi Truk Trailer'];
AnimTbuFigH = findobj(0, 'Name', AnimTbuFigTitle);
% ===== % No figure, initialize everything
if isempty(AnimTbuFigH),
    ui_row_n = 2;
    AnimTbuFigH = figure( ...
        'Name', AnimTbuFigTitle, ...
        'NumberTitle', 'off', ...
        'DockControls', 'off');
    figPos = get(AnimTbuFigH, 'position');
    % ===== proportion of UI frame and axes
    ui_area = 0.07;
    axis_area = 1-ui_area;
    % ===== animation area
    axisPos = [0 figPos(4)*ui_area figPos(3)
figPos(4)*axis_area];
    axisPos = axisPos./[figPos(3) figPos(4) figPos(3)
figPos(4)];
    axisPos = [0 ui_area 1 axis_area];

    axisPos = [0.1300 0.1100 0.7750 0.8150];
    axisPos = [0 0.1 1 0.8];
    AnimTbuAxisH = ...
        axes('unit', 'normal', 'pos', axisPos, 'visible', 'on');

% ##### animation objects #####
% ===== truck body
truck_l = 0.3; % This is defined in tbuinit.m
truck_w = 0.1;
TbuTruck = truck_l/2*[-1 1 1 1 1 1 1 -1 -1]' + ...
    sqrt(-1)*truck_w/2*[-1 -1 0 0 0 0 1 1 -1]';
truckH = line(real(TbuTruck), imag(TbuTruck));

trail_l = 0.15; % This is defined in tbuinit.m
trail_w = 0.1;
TbuTrail = trail_l/2*[-1 0.8 0.8 1 1 0.8 0.8 -1 -1]' + ...
    sqrt(-1)*trail_w/2*[-1 -1 -0.1 -0.1 0.1 0.1 1 1 -1]';

trail1H = line(real(TbuTrail), imag(TbuTrail));
trail2H = line(real(TbuTrail), imag(TbuTrail));
trail3H = line(real(TbuTrail), imag(TbuTrail));

```

```

    set(truckH, 'erasemode', 'normal','color', [0 1
0]),'userdata',TbuTruck);
    set(trail1H, 'erasemode', 'none','color', [0 0
1]),'userdata',TbuTrail);
    set(trail2H, 'erasemode', 'none','color', [0 1
1]),'userdata',TbuTrail);
    set(trail3H, 'erasemode', 'none','color', [1 0
1]),'userdata',TbuTrail);
    set(AnimTbuFigH,
'userdata',[truckH;trail1H;trail2H;trail3H]);

% ===== axis settings
max_x = 3; max_y = 5;
set(AnimTbuAxisH, 'clim', [1 64], ...
    'xlim', [-max_x max_x], ...
    'ylim', [-1 max_y], ...
    'box', 'on');
axis square;
% set(AnimTbuAxisH, 'visible', 'off');
% ===== plot back wall and dock
% x = [-max_x-5 -truck_w*0.8 0 truck_w*0.8 max_x+5];
% y = [0 0 nan 0 0];
% line(x, y, 'linewidth', 2, 'color', 'b');
% line([-truck_w -truck_w truck_w truck_w]*0.8, ...
% [1 -0.5 -0.5 1], 'linewidth', 3, 'color', 'black');
% x = [-2 2 2 2];
% y = [2 2 2 4];
% line(x, y, 'linewidth', 2, 'color', 'k');
grid on;
%set(AnimTbuFigH, 'color', get(AnimTbuFigH, 'color'));

end
sys = [0 0 1 12 0 0]; %jumlah output
x0=[];
set(AnimTbuFigH, 'HandleVisibility', 'on');
elseif nargin == 5, % for callbacks of GUI
% tmp = get(AnimTbuFigH, 'userdata');
% dispmodeH = tmp(2, 1);
% objectH = tmp(3, 1:4);
% dispmode = get(dispmodeH, 'value');
% if dispmode == 0, % xor
% set(objectH, 'erasemode', 'xor');
% else
% set(objectH, 'erasemode', 'none');
% end
end
end

```