



**KENDALI PERGERAKAN ROBOT SEPAK BOLA DENGAN PID**

*(Propotional Integral Derivatif) KONTROL*

**TUGAS AKHIR**

**Program Studi**

**S1 Sistem Komputer**

**Oleh :**

**AFIF WAHYU PRASTYAWAN**

**13.41020.0083**

**INSTITUT BISNIS  
DAN INFORMATIKA**

**stikom**  
**SURABAYA**

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

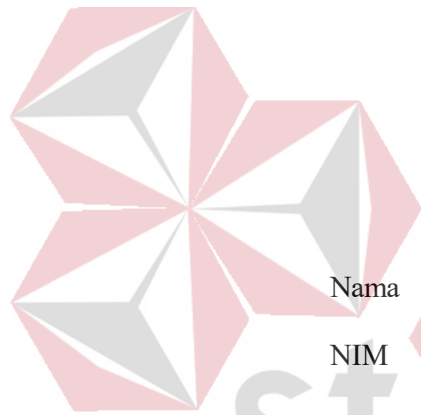
**INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

**2017**

**KENDALI PERGERAKAN ROBOT SEPAK BOLA DENGAN PID  
KONTROL**

**TUGAS AKHIR**

Diajukan sebagai salah satu syarat untuk menyelesaikan  
Program Sarjana Komputer



Disusun Oleh : **INSTITUT BISNIS**

Nama : Afif wahyu prastyawan **INFORMATIKA**

NIM : 13.41020.0083

Program : S1 (Strata Satu)

Jurusan : Sistem Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

**2017**

*'Kerja Keras, Kerja Cerdas, Kerja Ikhlas' Inshaallah hasil tidak akan  
menghianati proses*



INSTITUT BISNIS  
DAN INFORMATIKA

stikom  
SURABAYA

Syukur Alhamdulillah, Segala Puji Bagi ALLAH SWT Shalawat dan salam tidak

lupa selalu tercurahkan kepada Baginda Rasulullah SAW. Akhirnya penulis dapat

menyelesaikan Tugas Akhir ini

Tugas Akhir ini saya persembahkan kepada

Ibu, Bapak, Adik dan semua Keluarga yang selalu memberi dukungan dan

mendoakan saya

Terimakasih Kepada Dosen-Dosen Pembimbing

Serta para dosen yang selalu membimbing dan memberikan motivasi kepada saya.

Untuk Seluruh Rekan-Rekan di Sistem Komputer dan dikampus Institut Bisnis

dan Informatika Stikom Surabaya yang selalu membantu dan memberikan

motivasi kepada saya.

Beserta semua orang yang telah membantu.



**TUGAS AKHIR**  
**KENDALI PERGERAKAN ROBOT SEPAK BOLA DENGAN PID**  
**KONTROL**

Dipersiapkan dan disusun oleh  
**Afif Wahyu Prastyawan**  
NIM : 13.41020.0083

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji  
Pada : Februari 2017

Susunan Dewan Penguji

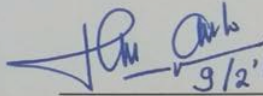
Pembimbing

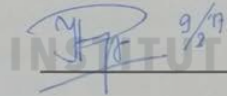
I. Harianto, S.Kom., M.Eng.  
NIDN. 0722087701

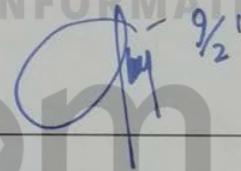
II. Ira Puspasari, S.Si., M.T  
NIDN. 0710078601

Pembahas

I. Dr. Jusak  
NIDN. 0708017101

  
9/2'17

  
9/2'17

  
9/2'17

Tugas Akhir ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Sarjana



FAKULTAS TEKNOLOGI  
DAN INFORMATIKA

**stikom**  
SURABAYA

Dr. Jusak

Dekan Fakultas Teknologi dan Informatika

**INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

## PERNYATAAN

### PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Afif wahyu prastyawan

NIM : 13410200083

Program Studi : S1 Sistem Komputer

Fakultas : Fakultas Teknologi dan Informatika

Jenis Karya : Tugas Akhir

Judul Karya : **KENDALI PERGERAKAN ROBOT SEPAK BOLA  
DENGAN PID KONTROL**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar keserjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, Februari 2017



Yang menyatakan

Afif Wahyu Prastyawan

Nim : 13410200083

## ABSTRAKSI

Salah satu contoh *mobile robot* yaitu, *mobile robot object tracking*. Dimana robot tersebut mempunyai cara kerja yaitu, robot akan mengikuti suatu benda tertentu sesuai yang diperintahkan. Untuk mengendalikan itu pengendalian sebuah *mobile robot*, maka diperlukan pengaturan posisi dan kecepatan motor yang baik. Sehingga diperlukan kendali yang berfungsi untuk mengendalikan posisi dan kecepatan motor DC. *Mobile robot* ini diterapkan sistem kontrol kendali cerdas pada *plant* yaitu kendali pergerakan pada *Mobile robot* yang berbasis PID (*Propotional Integral Derivatif*).

Dari hasil penelitian yang telah dibuat, didapatkan hasil respon pergerakan robot dari berbagai nilai variasi nilai  $K_p$ ,  $K_i$ ,  $K_d$  pada koordinat X awal 60 nilai *rise time* dan nilai *settling time* tercepat yaitu 4.7 detik dan 1.49 detik yang dihasilkan dari nilai  $K_p = 5.8$ ,  $K_i = 9$ ,  $K_d = 4.9$ , nilai *overshoot* paling rendah yaitu 22.74% dihasilkan dari variasi nilai  $K_p = 0.8$ ,  $K_i = 1$ ,  $K_d = 0.08$ . Pada koordinat X awal 330 nilai *rise time*, *settling time* tercepat dan nilai *overshoot* paling rendah yaitu 0.56 detik, 1.80 detik dan 14.28% dihasilkan dari variasi nilai  $K_p = 0.8$ ,  $K_i = 1$ ,  $K_d = 0.08$ . Pada koordinat X awal 560 nilai *rise time* dan nilai *settling time* tercepat yaitu 0.47 detik dan 1.47 detik dihasilkan dari nilai  $K_p = 2.8$ ,  $K_i = 5$  dan  $K_d = 0.8$ , nilai *overshoot* paling rendah yaitu 25.28% dihasilkan dari variasi nilai  $K_p = 0.8$ ,  $K_i = 1$ ,  $K_d = 0.08$ .

Kata kunci : *Mobile robot, tracking, PID (Proporsional-Integral- Derivatif)*.

## KATA PENGANTAR

Pertama-tama penulis panjatkan puji dan syukur atas kehadiran Allah SWT, karena berkat izin, Rahmat dan hidayahNya penulis dapat menyelesaikan laporan penelitian ini yang merupakan salah satu syarat Menempuh Tugas Akhir pada Program Studi S1 Sistem Komputer di Institut Bisnis dan Informatika Stikom Surabaya. Shalawat dan salam tidak lupa selalu penulis panjatkan kepada baginda Rasulullah SAW berkat beliau dunia berubah dari zaman kegelapan menuju ke zaman yang terang benderang.

Di dalam buku Laporan Tugas Akhir ini dilakukan pembahas mengenai pembuatan Kendali Pergerakan Robot Sepak Bola Dengan Pid Kontrol. Harapan penulis semoga laporan ini dapat berguna dan bermanfaat untuk menambah wawasan bagi pembacanya. Penulis juga menyadari dalam penulisan laporan Tugas Akhir ini masih terdapat banyak kekurangan.

Dalam usaha menyelesaikan Laporan Tugas Akhir ini penulis banyak mendapatkan bantuan dari berbagai pihak baik moral maupun materi. Oleh karena itu penulis mengucapkan terimakasih dan penghargaan setinggi-tingginya kepada:

1. Orang Tua dan Saudara-saudara saya tercinta yang telah memberikan dukungan dan bantuan baik moral maupun materi sehingga penulis dapat menempuh dan menyelesaikan Tugas Akhir maupun laporan ini.
2. Kepada Harianto, S.Kom, M.Eng. dan Ira Puspasari, S.Si., M.T. selaku Dosen Pembimbing. Terima kasih atas bimbingan yang diberikan sehingga penulis dapat menyelesaikan Tugas Akhir dengan baik.

3. Kepada Dr. Jusak. selaku Dosen penguji. Terima kasih atas bimbingan yang sehingga penulis dapat menyelesaikan Tugas Akhir dengan baik.
4. Kepada Anjik Sukmaaji, S.Kom., M.Eng selaku Ketua Program Studi Sistem Komputer Surabaya atas ijin yang diberikan untuk mengerjakan Tugas Akhir ini.
5. Semua staf Dosen yang telah mengajar dan memberikan ilmunya.
6. Terimakasih terhadap seluruh rekan-rekan S1 Sistem Kommputer khususnya Rekan-rekan seperjuangan angkatan 2013 khususnya Prodi S1 Sistem Komputer yang selalu memberikan semangat dan bantuannya.
7. Serta semua pihak lain yang tidak dapat disebutkan secara satu-persatu, yang telah membantu dalam menyelesaikan Tugas Akhir ini baik secara langsung maupun tidak langsung.

Surabaya, Februari 2017

Penulis

## DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
HALAMAN SYARAT.....	ii
MOTTO.....	iii
HALAMAN PERSEMBAHAN.....	iv
HALAMAN PENGESAHAN .....	v
HALAMAN PERNYATAAN .....	vi
ABSTRAK .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI .....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Perumusan Masalah .....	4
1.3 Batasan Masalah .....	4
1.4 Tujuan .....	5
1.5 Sistematika Penulisan .....	5
BAB II LANDASAN TEORI .....	7
2.1 Kotrol PID ( <i>Proporsional Integral Derivatif</i> ) .....	7
2.2 Mikrokontroler .....	10
2.3 Arduino.....	11

2.4	Arduino Mega 2560.....	11
2.4.1	Daya (power).....	14
2.4.2	Memori.....	15
2.4.3	Input dan output .....	15
2.5	<i>Software Arduino IDE</i> .....	16
2.6	Modul EMS 2A Dual H- Bridge.....	18
2.7	Modul Ems 5A H-Bridge .....	19
2.8	Motor DC 12V .....	21
2.9	Omni-Directional Wheels .....	22
2.10	Webcam Logitech C170 .....	24
BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM .....		26
3.1	Rancangan Penelitian.....	26
3.2	<i>Flowchart</i> Pergerakan Robot.....	28
3.3	Perancangan Mekanik Robot.....	29
3.4	Perancangan <i>Minimum System</i> .....	31
BAB IV HASIL DAN PEMBAHASAN.....		35
4.1	Pengujian Arduino .....	35
4.1.1.	Tujuan.....	35
4.1.2.	Alat yang digunakan.....	35
4.1.3.	Prosedur pengujian.....	35
4.1.4.	Hasil Pengujian.....	37
4.2	Pengujian Driver Motor .....	38
4.2.1.	Tujuan Pengujian Driver Motor .....	38
4.2.2.	Alat yang digunakan pada Pengujian <i>Driver</i> Motor .....	38

4.2.3.	Prosedur pengujian <i>Driver</i> Motor .....	39
4.2.4.	Hasil Pengujian <i>driver</i> Motor .....	40
4.3	Pengujian Sinkronisasi Pembacaan Koordinat Sensor Kamera.....	40
4.3.1.	Tujuan Pengujian Sinkronisasi Pembacaan Koordinat Sensor Kamera .....	41
4.3.2.	Alat Yang Digunakan Pada Pengujian Sinkronisasi Pembacaan Koordinat Sensor Kamera.....	41
4.3.3.	Prosedur Pengujian Sinkronisasi Pembacaan Koordinat Sensor Kamera .....	41
4.3.4.	Hasil Pengujian Sinkronisasi Pembacaan Koordinat Sensor Kamera .....	42
4.4	Pengujian Sistem.....	43
4.4.1.	Tujuan Pengujian Pengujian Sistem .....	43
4.4.2.	Alat Yang Digunakan Pada Pengujian Sistem.....	43
4.4.3.	Prosedur Pengujian Sistem.....	43
4.4.4.	Hasil Pengujian sistem.....	44
BAB V	PENUTUP .....	61
5.1	Kesimpulan .....	61
5.1	Saran .....	62
DAFTAR	PUSTAKA .....	63
LAMPIRAN	.....	65
BIODATA	PENULIS.....	80



## DAFTAR TABEL

	<b>Halaman</b>
Tabel 4.1 Hasil pengujian pergerakan <i>driver</i> motor .....	40
Tabel 4.2 Hasil pengujian sinkronisasi pembacaan koordinat sensorkamera	42
Tabel 4.3 Tabel pengujian pertama dengan nilai $K_p = 2,8$ , $K_i = 5$ dan $K_d = 0,8$ .....	44
Tabel 4.4 Tabel pengujian kedua dengan nilai $K_p = 2,8$ , $K_i = 5$ dan $K_d = 0,8$ .....	46
Tabel 4.5 Tabel pengujian ketiga dengan nilai $K_p = 2,8$ , $K_i = 5$ dan $K_d = 0,8$ .....	48
Tabel 4.6 Hasil respon dari pengujian pergerakan robot dengan nilai $K_p = 2,8$ , $K_i = 5$ , $K_d = 0,8$ .....	49
Tabel 4.7 Tabel pengujian pertama dengan nilai $K_p = 0,8$ , $K_i = 1$ , $K_d = 0,08$ .....	50
Tabel 4.8 Tabel pengujian kedua dengan nilai $K_p = 0,8$ , $K_i = 1$ , $K_d = 0,08$ .....	52
Tabel 4.9 Tabel pengujian ketiga dengan nilai $K_p = 0,8$ , $K_i = 1$ , $K_d = 0,08$ .....	54
Tabel 4.10 Hasil respon dari pengujian pergerakan robot dengan nilai $K_p = 0,8$ , $K_i = 1$ , $K_d = 0,08$ .....	55
Tabel 4.11 Tabel pengujian pertama dengan nilai $K_p = 5,8$ , $K_i = 9$ , $K_d = 4,9$ .....	57

Tabel 4.12	Tabel pengujian kedua dengan nilai $K_p = 5.8$ , $K_i = 9$ , $K_d = 4.9$ .....	58
Tabel 4.13	Tabel pengujian ketiga dengan nilai $K_p = 5.8$ , $K_i = 9$ , $K_d = 4.9$ .....	60
Tabel 4.14	Hasil respon dari pengujian pergerakan robot dengan nilai $K_p = 5.8$ , $K_i = 9$ , $K_d = 4.9$ .....	62



## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Blok Diagram PID .....	8
Gambar 2.2 Arduino Mega 2560 Sisi Depan (Kiri) dan Belakang (Kanan).	13
Gambar 2.3 Tampilan <i>Software</i> Arduino IDE.....	18
Gambar 2.4 Modul EMS 2A Dual H-Bridge .....	18
Gambar 2.5 Modul EMS 5A H-Bridge .....	20
Gambar 2.6 Motor DC.....	22
Gambar 2.7 Roda Omni (kiri), bentuk skematik robot three omni <i>directional</i> (kanan).....	23
Gambar 2.8 Webcam Logitech C170.....	25
Gambar 3.1 Diagram perangkat keras penelitian.....	26
Gambar 3.2 Diagram perancangan sistem.....	27
Gambar 3.3 <i>Flowchart</i> pergerakan robot .....	28
Gambar 3.4 Gambar desain robot.....	29
Gambar 3.5 Rangkaian Skematik Minimum Sistem.....	32
Gambar 4.1 <i>Upload program</i> berhasil .....	37
Gambar 4.2 <i>Program</i> berhasil berjalan.....	38
Gambar 4.3 Hasil plotting pengujian pergerakan robot menurut koordinat X.	46
Gambar 4.4 Hasil plotting pengujian pergerakan robot menurut koordinat X.	47
Gambar 4.5 Hasil plotting pengujian pergerakan robot menurut koordinat X	49
Gambar 4.6 Hasil plotting pengujian pergerakan robot menurut koordinat X	52

Gambar 4.7	Hasil plotting pengujian pergerakan robot menurut koordinat X	53
Gambar 4.8	Hasil plotting pengujian pergerakan robot menurut koordinat X	55
Gambar 4.9	Hasil plotting pengujian pergerakan robot menurut koordinat X	58
Gambar 4.10	Hasil plotting pengujian pergerakan robot menurut koordinat X	60
Gambar 4.11	Hasil plotting pengujian pergerakan robot menurut koordinat X	61



## BAB 1

### PENDAHULUAN

#### 1.1. Latar Belakang

Bidang robotik merupakan bagian dari ilmu komputer yang tidak terpisahkan. Implementasi robotik selama ini telah dapat membantu manusia di berbagai bidang terutama bidang industri. Robot dapat menggantikan manusia untuk melakukan berbagai hal yang tidak dapat dilakukan manusia.

Tentu saja semua ini tidak terlepas dari peran AI (*Artificial Intelligence*) yang menggerakkan robot-robot tersebut. AI memegang peran penting dalam dunia robotik karena memungkinkan robot dapat bergerak secara otomatis hanya dengan perintah sederhana. Bahkan untuk AI yang lebih canggih, campur tangan manusia tidak diperlukan.

Salah satu robot yang sering digunakan adalah *mobile robot*. *Mobile robot* adalah robot yang mempunyai aktuator berupa roda yang berfungsi untuk menggerakkan seluruh badan robot menuju ke suatu titik tertentu. *Mobile robot* salah satunya bisa digunakan untuk *tracking* (pelacakan), bisa *tracking* berupa benda, warna, garis, dan lain-lain

Salah satu contoh *mobile robot* yaitu, *mobile robot object tracking*. Dimana robot tersebut mempunyai cara kerja yaitu, robot akan mengikuti suatu benda tertentu sesuai yang diperintahkan. *mobile robot* ini tak lepas dari penerapan sistem kontrol kendali cerdas pada *plant* yaitu kendali pergerakan pada *mobile robot* yang berbasis PID (*Propotional Integral Derivatif*). Karena itulah

kendali PID diperlukan disini yaitu untuk mengendalikan posisi dan kecepatan motor DC.

Perkembangan dalam bidang robotika di Indonesia semakin pesat yang semula hanya sebagian orang menekuni bidang tersebut sekarang semakin banyak orang meminati dan menekuni atau sekedar hanya mengikuti berita tentang robotika. Semakin banyak lomba yang diadakan di bidang robotika membuat dunia robot di Indonesia menjadi lebih berkembang seperti robot soccer. *mobile robot* merupakan konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain. Robot soccer merupakan autonomous robot khusus atau *mobile robot* yang digunakan untuk memainkan pertandingan sepak bola. Robot diberi kemampuan mendeteksi, mengejar dan menendang bola, sehingga robot perlu diprogram untuk dapat mencapai kemampuan yang maksimal, sehingga dibutuhkan simulasi agar robot dapat mendeteksi bola dan berjalan sesuai keinginan.

Pada robotika pengendalian posisi dan kecepatan motor DC juga sangat penting misalnya dalam Kontes Robot Indonesia (KRI) dan Kontes Robot Cerdas Indonesia (KRCI). Robot harus dapat bergerak cepat dan tepat, meskipun terdapat berbagai halangan ataupun gangguan. Karena itu pergerakan robot memerlukan pengaturan posisi dan kecepatan motor yang baik agar tujuan yang diinginkan dapat tercapai.

Robot dengan roda omni adalah sebuah robot yang sistem gerakannya menggunakan roda omni. Roda omni telah digunakan bertahun tahun dalam dunia

robot industri dan logistik. Sumber utama pengguna terbanyak roda omni adalah perusahaan yang memproduksi untuk sistem konveyor, seperti untuk menangani paket/barang. Roda omni banyak juga digunakan untuk robot omni. Sebuah robot omni dapat berjalan lurus dari titik A ke titik B juga berputar agar dapat tiba di tujuannya. Roda omni juga digunakan untuk kursi roda, kendaraan servis di bandara dan lain-lain. Desain dengan roda omni dapat berjalan kedepan dan kesamping tanpa mengubah posisi roda. Hal tersebut lah yang membuat robot dengan roda omni lebih unggul dibanding robot dengan roda biasa (Irfan, 2016).

Pada tahun 2014 dilakukan penelitian oleh Rochmanto dengan judul Implementasi Robot *Three omni-directional* Menggunakan Kontroler PID pada Robot Kontes Robot ABU Indonesia. Kontes Robot Abu Indonesia merupakan ajang perlombaan nasional yang diadakan setiap tahun. Salah satu tugas pada perlombaan ini adalah robot harus dapat memindahkan obyek pada titik – titik tertentu di lapangan perlombaan dengan cara yang cepat dan efisien. Tugas akhir ini merancang dan mengimplementasikan robot *three omni-directional* menggunakan kontroler PID pada robot KRAI. Kelebihan robot *three omni-directional* adalah dapat bergerak ke segala arah tanpa harus mengubah arah hadapnya. Arah gerak pada robot ini bergantung pada perbandingan kecepatan pada tiap roda yang didapat dari perhitungan kinematika robot. Kontroler PID digunakan untuk mengatur kecepatan roda sehingga didapat respon kecepatan yang cepat mencapai *set point* dan stabil (Rochmanto, 2014).

Pengendali PID merupakan pengendali yang umum digunakan dalam berbagai macam proses industri. Popularitas pengendali PID disebabkan

khususnya karena performansinya yang baik dalam jangkauan yang lebar dari berbagai kondisi operasi dan khususnya dalam kesederhanaan fungsi PID, yang memungkinkan engineer untuk mengoperasikannya secara sederhana dan langsung. Untuk mengimplementasikan pengendali PID, tiga parameter harus ditentukan pada proses yang dikendalikan yang meliputi proportional gain, integral gain, dan derivative gain.

Karena pada penelitian kali ini akan pengendalian sebuah *mobile robot*, maka memerlukan pengaturan posisi dan kecepatan motor yang baik, yaitu diperlukan kendali PID yang berfungsi untuk mengendalikan posisi dan kecepatan motor DC, karena robot harus dapat bergerak cepat dan tepat untuk mencapai objek atau tujuan yang diinginkan.

### **1.2. Perumusan Masalah**

Adapun permasalahan yang akan dihadapi oleh penulis ke depannya dalam proses pengerjaan Tugas Akhir ini adalah:

1. Bagaimana sinkronisasi antara pembacaan koordinat dari *image processing* kamera saat diterapkan pada *mobile robot*.
2. Bagaimana merancang suatu pengendali motor DC dengan kendali PID.

### **1.3. Pembatasan Masalah**

Dalam perancangan dan pembuatan alat ini, terdapat beberapa batasan masalah, antara lain:

1. Menggunakan roda omni.
2. Dimensi robot yang digunakan sama dengan robot sepak bola beroda (ESRB).



3. Program deteksi dan *tracking* bola yang diterapkan adalah penelitian (putra, 2016).
4. Bola yang digunakan sebagai objek adalah bola tenis.
5. Posisi robot dengan bola tidak terdapat halangan.

#### **1.4. Tujuan**

Adapun tujuan dari pembuatan kendali *mobile robot* adalah sebagai berikut:

1. Sinkronisasi antara pembacaan koordinat dari image processing kamera saat diterapkan pada *mobile robot*.
2. Merancang suatu pengendali motor DC dengan kendali PID.

#### **1.5 Sistematika Penulisan**

Laporan Tugas Akhir ini ditulis dengan sistematika penulisan sebagai berikut:

##### **BAB I : PENDAHULUAN**

Bab ini membahas tentang latar belakang masalah, perumusan masalah, pembatasan masalah, tujuan penulisan laporan tugas akhir, dan sistematika penulisan tugas akhir.

##### **BAB II : LANDASAN TEORI**

Bab ini melakukan pembahasan tentang teori penunjang yang digunakan sebagai acuan dalam pengerjaan Tugas Akhir.

### **BAB III : METODE PENELITIAN**

Dalam bab ini dijelaskan tentang metode penelitian serta alasan penggunaan metode tersebut dalam penelitian. Pada bab ini dijelaskan pula tentang pembuatan perangkat keras (*hardware*) dengan menggabungkan perangkat lunak (*software*) sebagai pengontrol pada alat tersebut, serta penerapan metode penelitian pada alat ini.

### **BAB IV : PENGUJIAN DAN ANALISIS SISTEM**

Bab ini berisi tentang pengujian secara keseluruhan. Pengujian yang dilakukan meliputi pengujian error saat mencari arah yang dituju, error saat *mobile robot* sampai ke tujuan dan pengujian secara keseluruhan *mobile robot* berhasil sampai ke arah yang dituju.

### **BAB V : PENUTUP**

Bab ini berisi tentang kesimpulan penelitian serta saran untuk pengembangan penelitian.

INSTITUT BISNIS  
DAN INFORMATIKA

stikom  
SURABAYA

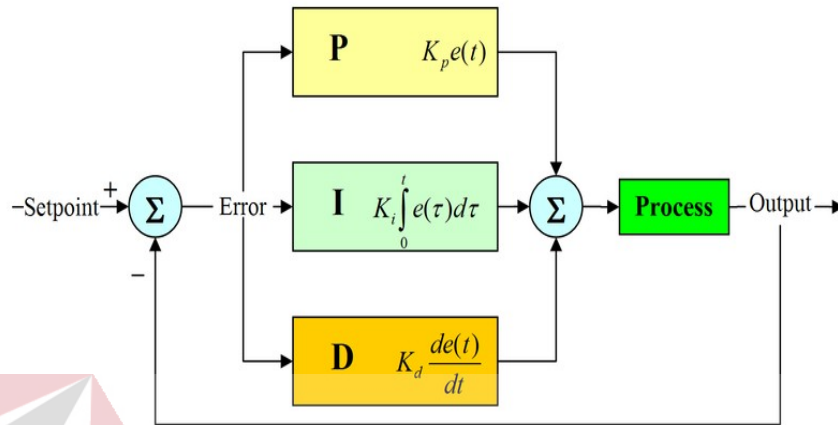
## BAB II

### TINJAUAN TEORI

#### 2.1. Kotrol PID (*Proporsional Integral Derivatif* )

Pengendali PID (Proporsional Integral Derivatif), merupakan gabungan dari tiga sistem kendali yang bertujuan untuk mendapatkan keluaran dengan risetime yang tinggi dan galat yang kecil. Seperti yang kita ketahui bahwa sistem kendali Proporsional memiliki keunggulan yaitu risetime yang cepat tetapi sangat rentan dengan *overshot/undershot*, sistem kendali integral memiliki keunggulan untuk meredam galat, sedangkan sistem kendali derivative memiliki keunggulan memperkecil delta *error* atau meredam *overshot/undershot*. PID berdasarkan implementasinya dibedakan menjadi 2 yaitu analog dan digital, PID analog diimplementasikan dengan komponen elektronika yang berupa *resistor*, *capacitor*, dan *operational amplifier*, sedangkan PID digital diimplementasikan secara program (Mas'ud, Hafidz, Harianto, & Wibowo, 2012).

Ketika membuat suatu sistem kontrol, tentunya ada suatu hal yang menjadi tujuan. Tujuan dari pengontrolan suatu sistem tersebut biasa disebut dengan *set point*. sistem akan terus berusaha agar nilai dari *set point* tersebut selalu terpenuhi. Simpangan nilai antara *set point* dengan nilai kenyataan disebut dengan *error*.



Gambar 2.1. Blok Diagram PID

Dalam membuat suatu sistem kontrol yang ditanamkan pada robot tentunya harus diimplementasikan kedalam suatu sintak program. Suatu rumus nantinya akan sedikit berubah penampilannya jika telah diimplementasikan ke dalam program, karena pada pemrograman, hanya diperlukan sifat-sifat dan algoritma dari rumus tersebut. Pada perancangan sistem PID ini, akan dijabarkan mengenai pengimplementasian rumus menjadi suatu sintak program sebagai berikut :

a) Proposional

Rumus dasar dari kontroler proposional adalah :

$$\mathbf{P} \quad K_p e(t)$$

Dengan:

P adalah Proposional

$K_p$  adalah nilai konstanta proposional

$e(t)$  adalah nilai eror yang selalu akan berubah

Seperti yang dijelaskan pada dasar teori PID, bahwa nilai error didapat dari:

$$e(t) = Sp - Pv$$

Dengan;

Sp adalah *Set point*

Pv adalah Process Variable

b) Integral

Rumus dasar dari kontroler Integral adalah:

$$I \quad K_i \int_0^t e(\tau) d\tau$$

Dengan:

I adalah Integral

Ki adalah nilai konstanta integral

e(t) adalah nilai error

dt adalah nilai perubahan waktu (*second*)

Kontroler integral tidak dapat digunakan sendiri, maka dari itu harus digabungkan dengan kontroler proposional.

c) Derivativ

Rumus dasar dari kontroler Derivativ adalah:

$$D \quad K_d \frac{de(t)}{dt}$$

Dengan:

D adalah Derivativ

Kd adalah nilai konstanta Derivativ

de(t) adalah nilai perubahan error

dt adalah nilai perubahan waktu (*second*)

Sama juga halnya dengan kontroler Integral, kontroler Derivativ juga tidak bisa digunakan sendirian, melainkan juga harus digabungkan dengan kontroler Proposional. Apabila tidak, maka kontroler Derivativ tidak akan dapat melakukan aksi apapun, karena tidak ada perubahan error (Mas Pratama, Suwedan, & Swamardika, 2013).

## 2.2. Mikrokontroler

Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program didalamnya. Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program did umumnya terdiri dari CPU (*Central Processing Unit*), memori, I/O tertentu dan unit pendukung seperti *Analog-to-Digital Converter* (ADC) yang sudah terintegrasi di dalamnya. Kelebihan utama dari mikrokontroler ialah tersedianya RAM dan peralatan I/O pendukung sehingga ukuran board mikrokontroler menjadi sangat ringkas. Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program did MCS51 ialah mikrokomputer CMOS 8 bit dengan 4 KB Flash PEROM (*Programmable and Erasable Only Memory*) yang dapat dihapus dan ditulisi sebanyak 1000 kali.

Mikrokontroler ini diproduksi dengan menggunakan teknologi *high density nonvolatile memory*. Flash PEROM on-chip tersebut memungkinkan memori program untuk diprogram ulang dalam sistem (*in-system programming*) atau dengan menggunakan *programmer nonvolatile memory* konvensional. Kombinasi CPU 8 bit serba guna dan *Flash PEROM*, menjadikan mikrokontroler MCS51 menjadi *microcomputer* handal yang fleksibel.

Mikrokontroler tersusun dalam satu chip dimana prosesor, memori, dan I/O terintegrasi menjadi satu kesatuan kontrol sistem sehingga mikrokontroler dapat dikatakan sebagai komputer mini yang dapat bekerja secara inovatif sesuai dengan kebutuhan sistem.

Mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data (Naziq, tidak ada tahun).

### 2.3. Arduino

Arduino adalah *prototipe platform elektroonik opensource* yang terdiri mikrokontroler, bahasa pemrograman, dan IDE. Arduino adalah alat untuk membuat aplikasi interaktif, yang dirancang untuk mempermudah proyek bagi pemula, tapi masih fleksibel bagi para ahli untuk mengembangkan proyek – proyek yang kompleks (Arduino.cc)

## 2.4. Arduino Mega 2560

Arduino mega 2560 adalah papan mikrokontroler berdasarkan ATmega2560 (*datasheet*). Ini memiliki 54 digital pin input / output ( yang 15 dapat digunakan sebagai output PWM ) , 16 analog input , 4 UART ( hardware port serial ) , osilator kristal 16 MHz , koneksi USB , jack listrik , header ICSP , dan tombol reset. Semuanya diperlukan untuk mendukung kerja

mikrokontroler, cara mengaktifkan Arduino mega 2560 adalah dengan menghubungkannya ke komputer dengan kabel USB atau memberikan *power* dengan adaptor AC -DC atau baterai. Arduino Mega ini *compatible* dengan Arduino Duemilanove atau Diecimila.

Mega 2560 adalah update dari Arduino Mega. Mega2560 berbeda dari semua *board* sebelumnya yang tidak menggunakan FTDI chip driver USB -to -serial. Revisi ke 2 dari *board* Mega 2560 memiliki resistor 8U2, sehingga lebih mudah untuk dimasukkan ke dalam mode DFU .

Revisi 3 dari *board* Mega 2560 memiliki fitur- fitur baru berikut :

a. Pinout : menambahkan SDA dan pin SCL yang dekat dengan pin AREF dan dua pin baru lainnya ditempatkan dekat dengan pin RESET, IOREF yang memungkinkan *shield* untuk beradaptasi dengan tegangan yang tersedia dari *board*. Nantinya , *shield* akan *compatible* baik dengan *board* yang menggunakan AVR , yang beroperasi dengan 5V dan dengan Arduino Due yang beroperasi dengan 3.3V. Yang kedua adalah pin tidak terhubung , yang disediakan untuk tujuan lainnya.

b. Sirkuit RESET kuat.



c. Atmega 16U2 menggantikan 8U. (Arduino.cc)



Gambar 2.2 Arduino Mega 2560 Sisi Depan (Kiri) dan Belakang (Kanan).

Sumber : (Arduino.cc)

Secara umum Arduino terdiri dari dua bagian, yaitu:

1. *Hardware*: papan input/output (I/O) .
2. *Software*: *software* Arduino meliputi IDE untuk menulis program, driver untuk koneksi dengan komputer, contoh program dan *library* untuk pengembangan program (Djuandi, 2011) .Spesifikasi Arduino Mega 2500 dapat dilihat pada Tabel 3.1.

Tabel 3.1 Spesifikasi Arduino Mega 2560.

Mikrokontroler	ATMega 2560
Tegangan Operasi	5V
<i>Input</i> tegangan(rekomendasi)	7 – 12V
<i>Input</i> tegangan (Maksimal)	6 – 20V
<i>Digital</i> I/O Pin	54 (15 pin PWM)
<i>Pin input Analog</i>	16
DC current per I/O Pin	40mA
Pin DC Current untuk 3.3V	50mA

Memori <i>flash</i>	256Kb, 8Kb digunakan untuk <i>bootloader</i>
SRAM	8Kb
EEPROM	4Kb
<i>Clock speed</i>	16Hz

#### 2.4.1. Daya (*Power*)

Arduino mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Eksternal ( non- USB ) daya dapat berasal baik dari adaptor AC -DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan plug 2.1mm pusat - positif ke colokan listrik *board*. Baterai dapat dimasukkan dalam Gnd dan Vin pin header dari konektor daya.

*Board* dapat beroperasi pada pasokan eksternal 6 sampai 20 volt. Jika tegangan dengan kurang dari 7V, tegangan pada *board* kemungkinan akan tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak *board*. Kisaran yang disarankan adalah 7 sampai 12 volt.

Pin listrik adalah sebagai berikut :

VIN : Tegangan *input* ke *board* Arduino ketika itu menggunakan sumber daya eksternal ( sebagai lawan 5 volt dari koneksi USB atau sumber daya diatur lain ).

Kita dapat memasok tegangan melalui pin ini.

5V : Pin *output* 5V diatur dari regulator di *board*. *Board* dapat diaktifkan dengan daya baik dari colokan listrik DC ( 7 - 12V ) , konektor USB (5V) , atau pin VIN

dari board (7-12V). Jika tegangan diberikan melalui 5V atau 3.3V melewati regulator, dan dapat merusak *board*, maka tidak disarankan.

3V3 : Sebuah pasokan 3,3 volt dihasilkan oleh regulator on-board yang dapat menarik arus maksimum 50 mA.

GND : Pin tanah.

IOREF : Pin pada *board* Arduino memberikan tegangan referensi saat mikrokontroler sedang beroperasi. Sebuah *shield* dikonfigurasi dengan benar agar dapat membaca pin tegangan IOREF dan memilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan pada *output* untuk bekerja dengan 5V atau 3.3V. (Arduino.cc)

#### 2.4.2. Memori

ATmega2560 memiliki 256 KB dari flash memory untuk menyimpan kode (8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan *library* EEPROM). (Arduino.cc)

#### 2.4.3. Input dan Output

Masing-masing dari 54 digital pin pada Mega dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi dari `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Mereka beroperasi pada tegangan 5V. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor *pull-up internal* yang (terputus secara *default*) dari 20-50 KOhms. Selain itu, beberapa pin memiliki fungsi khusus :

1. **Serial** : 0(RX) dan 1(TX); Serial 1 : 19(RX) dan 18(TX) ; Serial 2 : 17(RX) dan 16 (TX) ; Serial 3 : 15 (RX) dan 14(TX). Yang digunakan untuk

menerima (RX) dan mengirimkan (TX) TTL data serial. Pin 0 dan 1 juga

terhubung ke pin yang sesuai dari ATmega16U2 USB - to- TTL *chip* Serial.

2. **Eksternal Interupsi:** 2 (*interrupt* 0) , 3 (*interrupt* 1) , 18 (*interrupt* 5) , 19 (*interrupt* 4) , 20 (*interrupt* 3) , dan 21 (*interrupt* 2). Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, naik atau jatuh tepi, atau perubahan nilai. Lihat *AttachInterrupt*( ) fungsi untuk rincian.
3. **PWM:** 2-13 dan 44 sampai 46. Menyediakan 8 - bit PWM *output* dengan *analogWrite* ( ) *function*.
4. **SPI :** 50(MISO), 51(MOSI), 52(SCK), 53(SS). Pin ini mendukung komunikasi SPI menggunakan *library* SPI. Pin SPI juga pecah pada header ICSP, yang secara fisik kompatibel dengan Uno , Duemilanove dan Diecimila .
5. **LED :** 13. Ada *built -in* LED terhubung ke pin digital 13. Ketika pin dengan nilai TINGGI , LED menyala , ketika pin yang dipakai RENDAH , lampu akan mati.
6. **TWI :** 20(SDA) dan 21(SCL). Dukungan komunikasi TWI menggunakan *library* Wire. Yang perlu diperhatikan adalah bahwa pin ini tidak berada di lokasi yang sama dengan pin TWI pada Duemilanove atau Diecimila. Arduino Mega2560 memiliki 16 input analog , yang masing-masing menyediakan 10 bit resolusi (yaitu 1024 nilai yang berbeda ). Secara *default* mereka mengukur dari *ground* sampai 5 volt , meskipun mungkin untuk

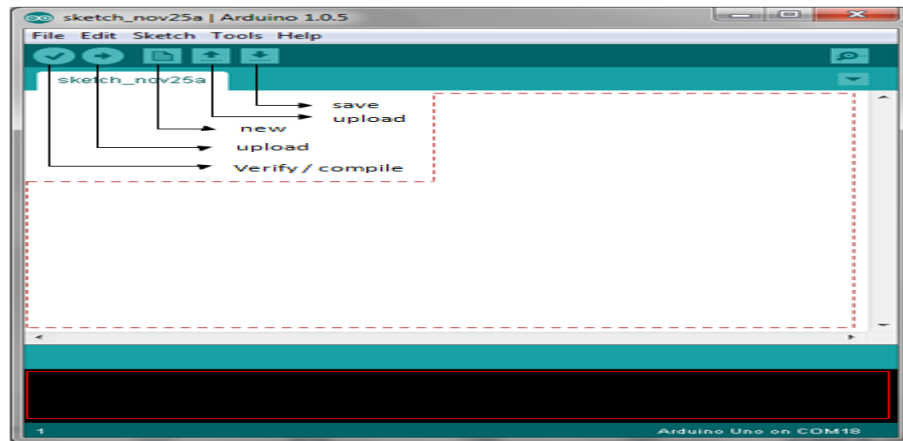
mengubah jangkauan menggunakan pin AREF dan fungsi `analogReference()` (Arduino.cc).

## 2.5. Software Arduino IDE

Arduino IDE adalah software yang ditulis menggunakan java dan berdasarkan pengolahan seperti, `avr-gcc`, dan perangkat lunak *open source* lainnya (Djuandi, 2011). Arduino IDE terdiri dari:

1. Editor program, sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.
2. *Verify / Compiler*, sebuah modul yang mengubah kode program (bahasa *processing*) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *processing*, yang dipahami oleh mikrokontroler adalah kode biner.
3. Pengunggah, sebuah modul yang memuat kode biner dari komputer ke dalam memori mikrokontroler di dalam papan Arduino.
4. Pada Gambar 8 terdapat *menu bar*, kemudian *toolbar* dibawahnya, dan sebuah area putih untuk *editing sketch*, area hitam dapat kita sebut sebagai *progress area*, dan paling bawah dapat kita sebut sebagai “*status bar*”.

Tampilan *Software* Arduino IDE ditunjukkan pada Gambar 2.3.



Gambar 2.3 Tampilan *Software* Arduino IDE

Sumber : (Arduino, 2015)

## 2.6. Modul EMS 2A Dual H-Bridge

EMS (*Embedded Module Series*) 2 A Dual H-Bridge merupakan *driver H-Bridge* yang didisain untuk menghasilkan *drive* dua arah dengan arus kontinu hingga 2 A pada tegangan 4,8 Volt sampai 46 Volt. Modul ini dapat men-*drive* beban-beban induktif seperti misalnya motor DC, motor *stepper*, koil relay, selenoida, dan beban-beban lainnya. Gambar dari Modul EMS 2A Dual H-Bridge dapat dilihat pada Gambar 2.4.



Gambar 2.4 Modul EMS 2A Dual H-Bridge

Spesifikasi :

1. Catu daya modul : 5VDC.
2. Terdiri dari 2 *driver full H-Bridge* yang dapat diparalel.
3. Tiap *driver* mampu melewatkan arus kontinu 2 A atau arus kontinu 4 A saat diparalel, serta mendukung tegangan beban dari 4,8 V sampai 46 V.
4. Input kompatibel dengan level tegangan TTL dan CMOS.
5. *Output tri-state* dan dilengkapi dengan dioda eksternal untuk pengamanan beban induktif.
6. Masing-masing *H-Bridge* dilengkapi sensor arus beban yang dapat digunakan sebagai umpan balik ke pengendali.
7. Proteksi *overtemperature*.
8. Jalur catu daya *input* (logika) terpisah dari jalur catu daya untuk beban.
9. Desain PCB standar industri dengan bahan 2 layer FR4 dan *plated through hole* (PTH).
10. Kompatibel penuh dengan SPC *Gamepad Interface*, serta mendukung sistem mikrokontroler.

## 2.7. Modul Ems 5A H-Bridge

Embedded Module Series (EMS) 5 A H-Bridge merupakan driver H-Bridge yang didisain untuk menghasilkan drive 2 arah dengan arus kontinyu sampai dengan 5 A pada tegangan 5 Volt sampai 40 Volt. Modul ini dilengkapi dengan rangkaian sensor arus beban yang dapat digunakan sebagai umpan balik ke pengendali. Modul ini mampu men-drive beban-beban induktif seperti misalnya relay, solenoida, motor DC, motor stepper, dan berbagai macam beban

lainnya. Gambar dari Modul EMS 2A Dual H-Bridge dapat dilihat pada Gambar 2.5.



Gambar 2.5 Modul EMS 5A H-Bridge

#### SPESIFIKASI

1. Terdiri dari 1 driver full H-Bridge beserta rangkaian *current sense*.
2. Mampu melewatkan arus kontinyu 5 A.
3. Range tegangan output untuk beban: 5 V sampai 40 V.
4. Input kompatibel dengan level tegangan TTL dan CMOS.
5. Jalur catu daya *input* (VCC) terpisah dari jalur catu daya untuk beban (V Mot).
6. *Output tri-state*.
7. Dilengkapi dengan dioda eksternal untuk pengaman beban induktif.
8. Frekuensi PWM sampai dengan 10 KHz.
9. *Active Current Limiting*.
10. Proteksi hubungan singkat.
11. Proteksi *overtemperature*.
12. *Undervoltage Shutdown*



## 2.8. Motor DC 12V

Motor adalah suatu mesin listrik yang mengubah energi listrik menjadi energi mekanik. Pada umumnya motor arus searah terdiri dari tiga bagian utama yaitu:

- Bagian yang diam disebut dengan *stator*
- Bagian yang berputar disebut dengan *rotor*
- *Komutator* dan sikat arang

*Stator* merupakan magnet permanen yang melekat pada lingkaran paling luar. Sedangkan *rotor* yang berhimpit dengan *stator*. *Komutator* ikut berputar dengan rotor yang berfungsi sebagai pengatur polaritas tegangan yang masuk ke rotor agar motor tetap berputar

Motor DC adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). (Elektronika T. , 2012). Motor DC pada umumnya berputar searah jarum jam, tetapi dapat juga berputar berlawanan arah jarum jam. Pemasangan yang tidak sesuai antara terminal positif dan negatif dapat menyebabkan motor berputar berlawanan. Contoh gambar motor DC dapat dilihat pada Gambar 2.6.



Gambar 2.6 Motor DC

## Spesifikasi:

1. *Voltage: 12.0VDC*
2. *Output Speed: 200 +/- 10% RPM*
3. *No-Load output current:  $\leq 50$  mA*
4. *Rotation Output: CW / CCW*
5. *Noise: No Gear Noise*
6. *Stall output: : Slip Gear, Broken Gear is no allowed*
7. *Output shaft of the axial clearance:  $\leq 0.1 \sim 0.3$ mm, Horizontal clearance requirement  $\leq 0.05$*

## 2.9. Omni-Directional Wheels

Roda Omni-directional adalah suatu roda unik karena memiliki kemampuan bergerak bebas dua arah. Roda ini berputar seperti roda pada umumnya serta mampu bergeser kesamping menggunakan roda di sepanjang lingkaran luar roda. Roda Omni-directional memungkinkan robot untuk mengkonversi dari robot nonholonomic untuk robot holonomic. Sebuah robot nonholonomic yang menggunakan roda normal hanya memiliki 2 DOF (Degree of

Freedom) yang terkendali, yaitu bergerak maju / mundur dan rotasi. Robot nonholonomic tidak memiliki kemampuan untuk bergerak kesamping kiri / kanan sehingga membuat robot lebih lambat dan kurang efisien dalam mencapai tujuan yang diberikan.

Roda omni-directional holonomic mampu mengatasi masalah ini karena roda memiliki 3 DOF. Berbeda dengan robot nonholonomic normal, robot omni-directional holonomic mampu bergerak ke segala arah tanpa mengubah arah roda. Roda omni-directional holonomic dapat bergerak maju mundur, geser ke samping, dan berputar pada posisi tetap. Kemampuan ini memungkinkan robot yang menggunakan omnidirectional mampu bermanuver untuk lebih lincah dan lebih efisien. Gambar Omni ditunjukkan pada Gambar 2.7.



Gambar 2.7 Roda Omni (kiri), bentuk skematik robot three omni  
*directional*(kanan)

Melalui Gambar 2.7 persamaan kinematik dari sistem pergerakan ini dapat diperoleh. Persamaan yang digunakan pada sistem kontrol robot adalah:

$$V_x = V_3 - V_1 \cos(\delta) - V_2 \cos(\delta) \dots\dots\dots(2-1)$$

$$V_y = V_1 \sin(\delta) - V_2 \sin(\delta) \dots\dots\dots(2-2)$$

$$V_\phi = V_1/L + V_2/L + V_3/L \dots\dots\dots(2-3)$$

$$V_{i(1,2,3)} = w \cdot r \dots\dots\dots(2-4)$$

Dengan :

v : Vektor putaran roda omni

r : jari-jari roda omni (cm)

w : kecepatan angular dari roda (rad/detik)

Jadi kecepatan pada tiap roda didapatkan dengan mengalikan antara kecepatan sudut roda dengan jari-jari roda secara analisis teori. Setiap roda disusun secara simetris dengan perbedaan sudut tiap roda ( $120^\circ$ ) dengan  $\delta = (60^\circ)$ . Jadi persamaan 2-1, 2-2, dan 2-3 dapat ditulis dengan suatu bentuk matrik (Rochmanto, 2014).

$$\begin{bmatrix} V_x \\ V_y \\ V_\phi \end{bmatrix} = \begin{bmatrix} -\cos\delta & -\cos\delta & 1 \\ \sin\delta & -\sin\delta & 0 \\ 1/L & 1/L & 1/L \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \dots\dots\dots(2-5)$$

## 2.10. Webcam Logitech C170

Kamera yang digunakan dalam penelitian ini adalah webcam Logitech C170. Bentuk fisik dari webcam Logitech C170 ditunjukkan pada Gambar 2.9.

Berikut spesifikasi sebagai dari webcam Logitech C170:

- a) Panggilan video (640 x 480 piksel) dengan sistem yang direkomendasikan
- b) Merekam video: Hingga 1024 x 768 piksel
- c) Teknologi Logitech Fluid Crystal™\*
- d) Foto: Hingga 5 megapiksel (ditingkatkan menggunakan perangkat lunak)

- e) Mikrofon bawaan dengan pengurangan *noise*.
- f) Bersertifikat *Hi-Speed* USB 2.0 (direkomendasikan)
- g) Klip universal cocok dengan berbagai laptop, monitor LCD atau CRT

Kebutuhan sistem dari webcam Logitech C170:

- h) 1 GHz (Direkomendasikan 1.6 GHz)
- i) RAM 512 MB atau lebih
- j) Kapasitas *hard drive* 200 MB
- k) Port USB 1.1 (direkomendasikan 2.0)



Gambar 2.8 Webcam Logitech C170

(Sumber : [www.logitech.com](http://www.logitech.com))

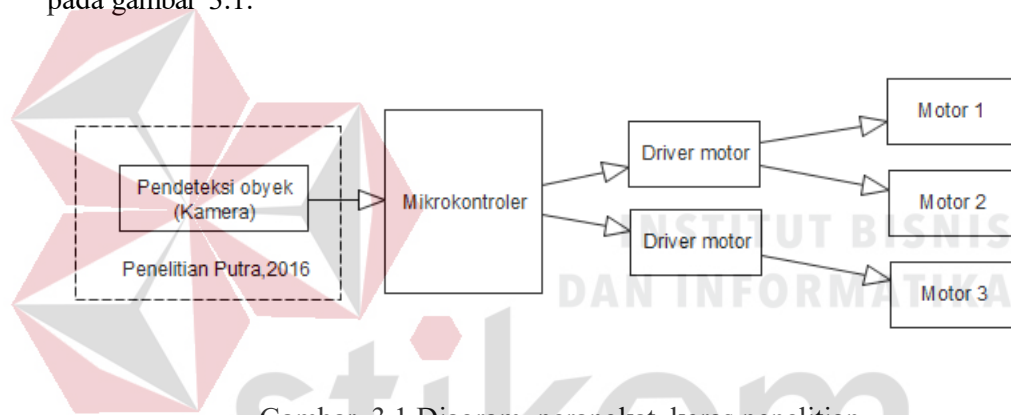
## BAB III

### METODE PENELITIAN

#### 3.1 Rancangan Penelitian

##### 1. Perancangan perangkat keras

Secara umum perancangan perangkat keras meliputi unit masukan dan keluaran pada mikrokontroler. Perancangan perangkat keras dapat dilihat pada gambar 3.1.



Gambar 3.1 Diagram perangkat keras penelitian

Tiap-tiap bagian dari diagram rangkaian alat pada Gambar 3.1 dapat dijelaskan sebagai berikut:

##### a. Input pada mikrokontroler:

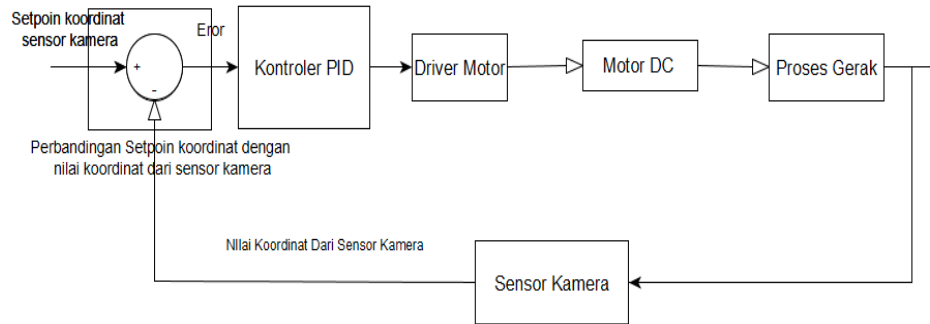
Kamera: Digunakan sebagai pendeteksi bola

##### b. Output pada mikrokontroler:

Driver motor: Digunakan untuk mengontrol arah putaran dan kecepatan motor DC

Motor DC: Sebagai penggerak mekanik robot

## 2. Perancangan sistem

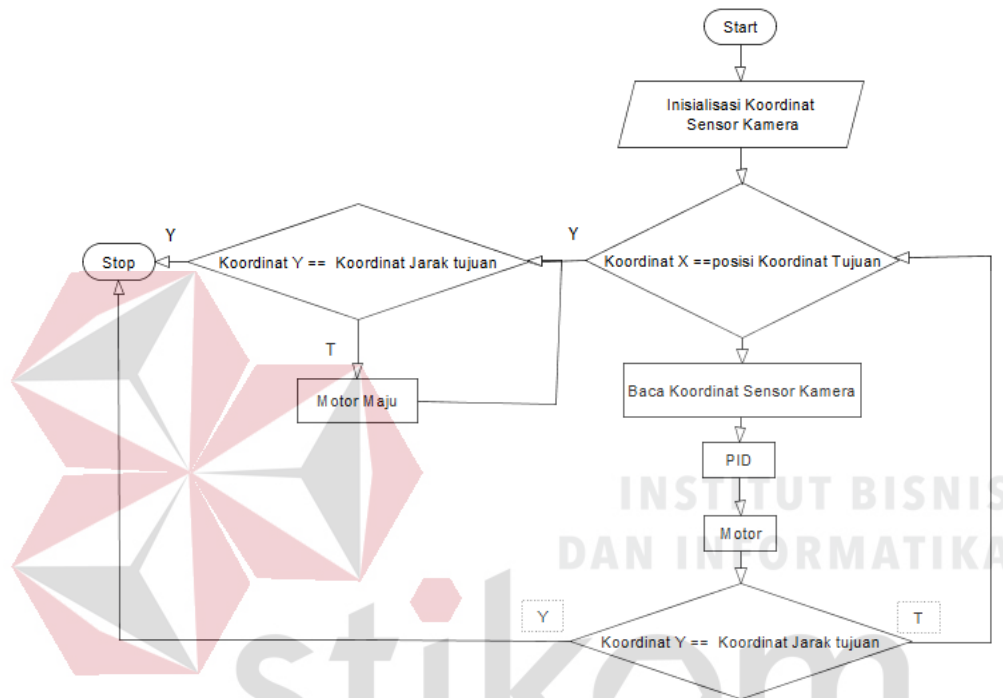


Gambar 3.2. Diagram perancangan sistem

*Set point* pada penelitian ini adalah suatu keadaan dimana robot diharuskan untuk tetap berada lurus dengan bola yaitu berada pada koordinat  $X=330$ . Nilai eror didapatkan dari hasil perbandingan *set point* koordinat dengan nilai koordinat dari sensor kamera saat itu. Kemudian diproses menggunakan kontroler PID. Sensor kamera yang berfungsi sebagai deteksi bola atau sebagai *feedback* dari kontrol PID tersebut diteliti oleh (Putra,2016).

### 3.2 Flowchart Pergerakan Robot

Untuk dapat menuju pada bola diperlukan beberapa tahapan seperti yang terlihat pada gambar 3.3.



Gambar 3.3 Flowchart pergerakan robot

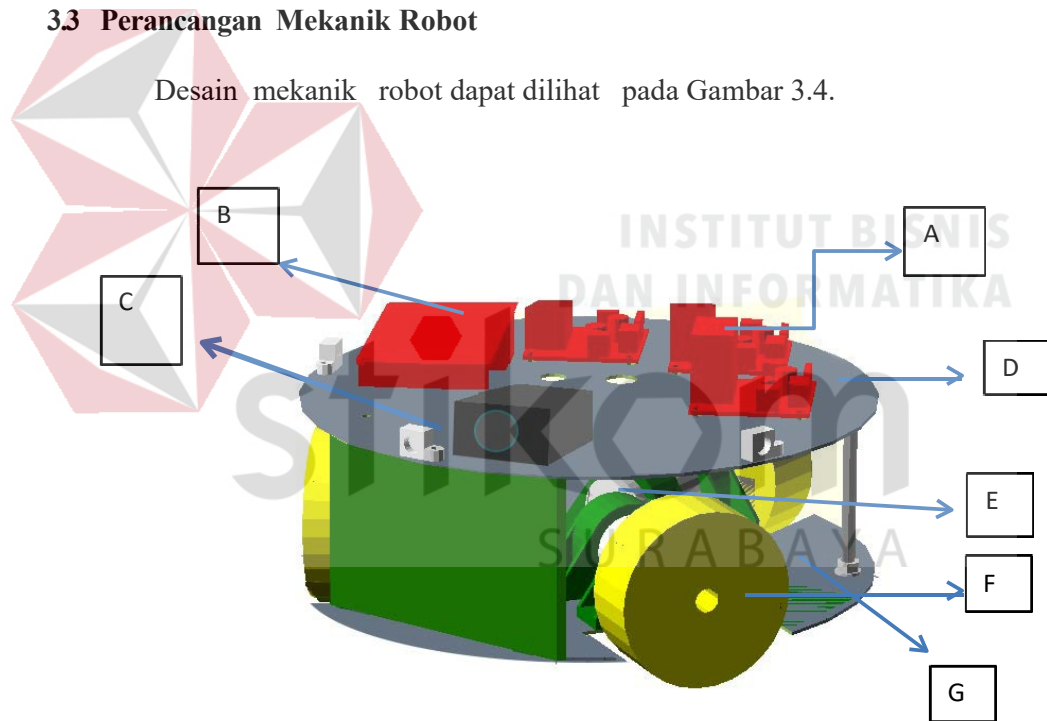
- Inisialisasi koordinat sensor kamera.
- Koordinat X yang dihasilkan dari sensor kamera akan dibandingkan dengan koordinat posisi yang diinginkan atau dengan posisi koordinat tujuan
- Jika Koordinat X belum sesuai dengan koordinat yang diinginkan, maka program akan terus membaca koordinat dari sensor kamera. Proses PID untuk memperbaiki arah gerak robot untuk menuju koordinat yang diinginkan.



- d. Ketika Koordinat X sudah terpenuhi atau sesuai dengan yang diinginkan, kemudian membandingkan koordinat Y dengan koordinat jarak tujuan. Jika koordinat Y belum sesuai dengan yang diinginkan maka robot akan terus berjalan sampai dengan koordinat Y yang diinginkan.
- e. Program akan berhenti jika Koordinat X dan koordinat Y sudah sesuai dengan yang diinginkan

### 3.3 Perancangan Mekanik Robot

Desain mekanik robot dapat dilihat pada Gambar 3.4.



Gambar 3.4 Gambar desain robot.

Keterangan:

Gambar A= Motor driver

Gambar B = Mikrokontroler

Gambar C = kamera

Gambar D = Base 2

Gambar E= Motor DC

Gambar F= Roda

Gambar G= Base 1

### **331. Ukuran Dimensi Robot**

Setelah semua komponen tambahan dari penelitian ini dipasangkan, terdapat ukuran dimensi dari keseluruhan robot :

1. Diameter Robot : 35 cm
2. Tinggi Robot : 12 cm

### **332. Struktur Material Robot**

Bahan material yang digunakan dalam penelitian ini menggunakan beberapa bahan diantaranya sebagai berikut :

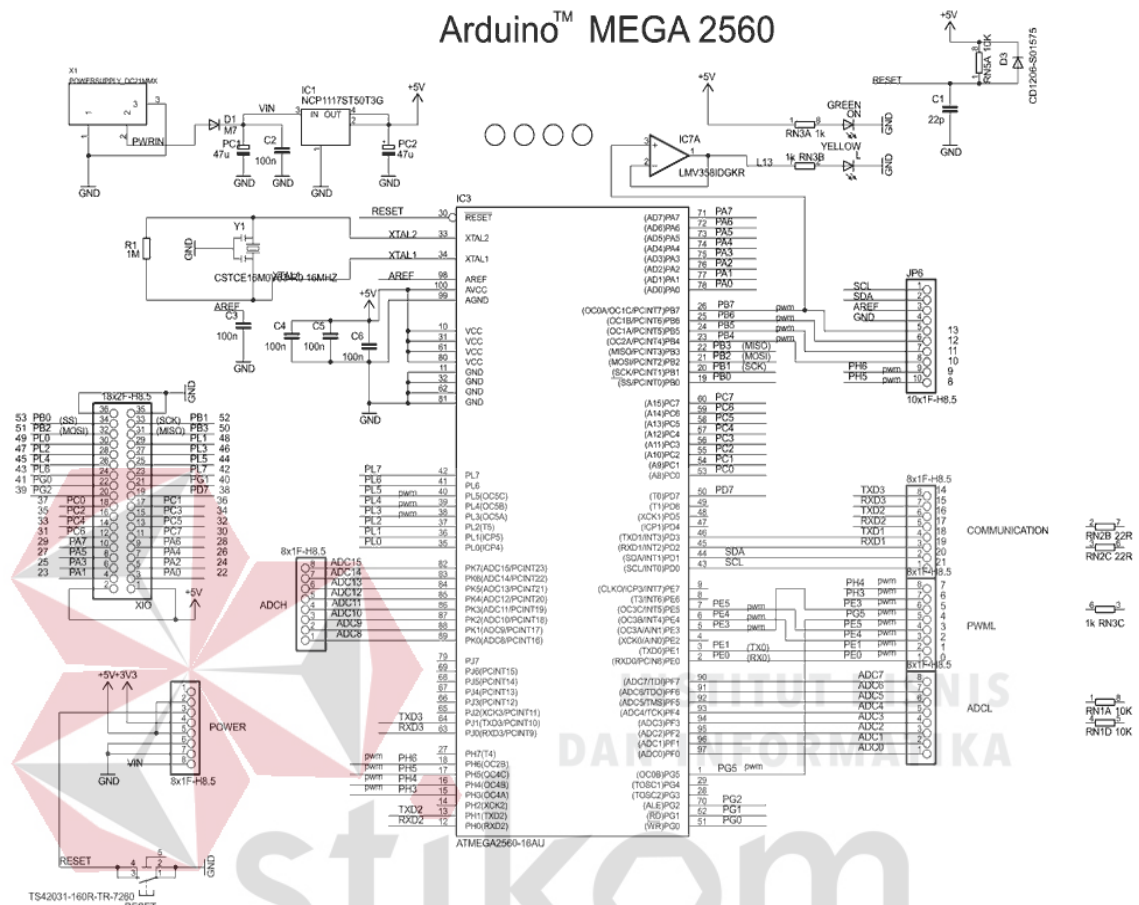
- a. Bagian rangka
  1. Aluminium
  2. Mur dan Baut
  3. Akrilik

b. Bagian dari Perangkat Robot

1. Motor DC 12 volt
2. Roda Omni
3. *Driver* motor

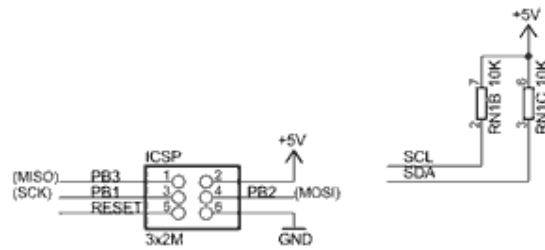
### 3.4 Perancangan *Minimum System*

Pada tugas akhir ini dibuat sebuah pengendali menggunakan mikrokontroler keluaran dari Arduino, yaitu Arduino Mega 2560. Untuk menjalankan mikrokontroler ini diperlukan rangkaian *minimum system*. Rangkaian *minimum system* tersebut terdiri rangkaian *reset* dan rangkaian osilator. Dalam perancangan *minimum system* memerlukan beberapa komponen pendukung seperti kristal, resistor dan kapasitor. Rangkaian *minimum system* dibuat untuk mendukung kerja dari mikrokontroler Arduino dimana mikrokontroler tidak bisa berdiri sendiri atau harus ada rangkaian dan komponen pendukung seperti halnya rangkaian catu daya, kristal dan lain sebagainya yang biasanya disebut *minimum system*. Pada penelitian ini menggunakan rancangan elektronik, seperti yang ditunjukkan pada Gambar 3.5.



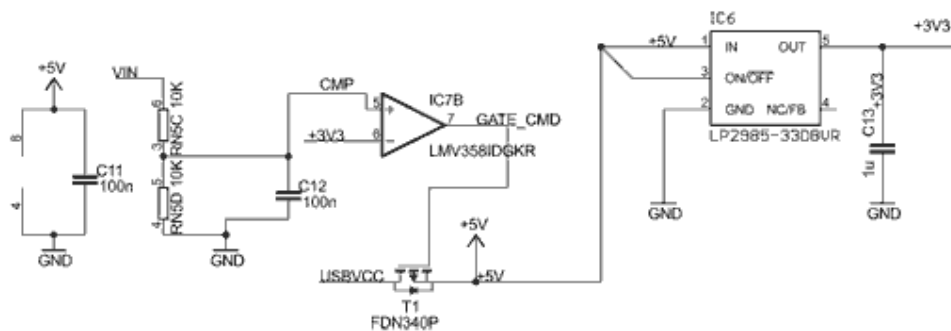
Gambar 3.5 A. Rangkaian Skematik ATMEGA2560 Arduino Mega

ATMEGA2560 yaitu sebuah *system microprocessor* yang berada di dalam mikrokontroler Arduino Mega, dimana didalamnya sudah terdapat CPU, ROM, RAM, I/O, *Clock* dan peralatan *internal* lainnya yang sudah saling terhubung dan terorganisasi dengan baik dan dikemas dalam satu *chip* yang siap pakai.



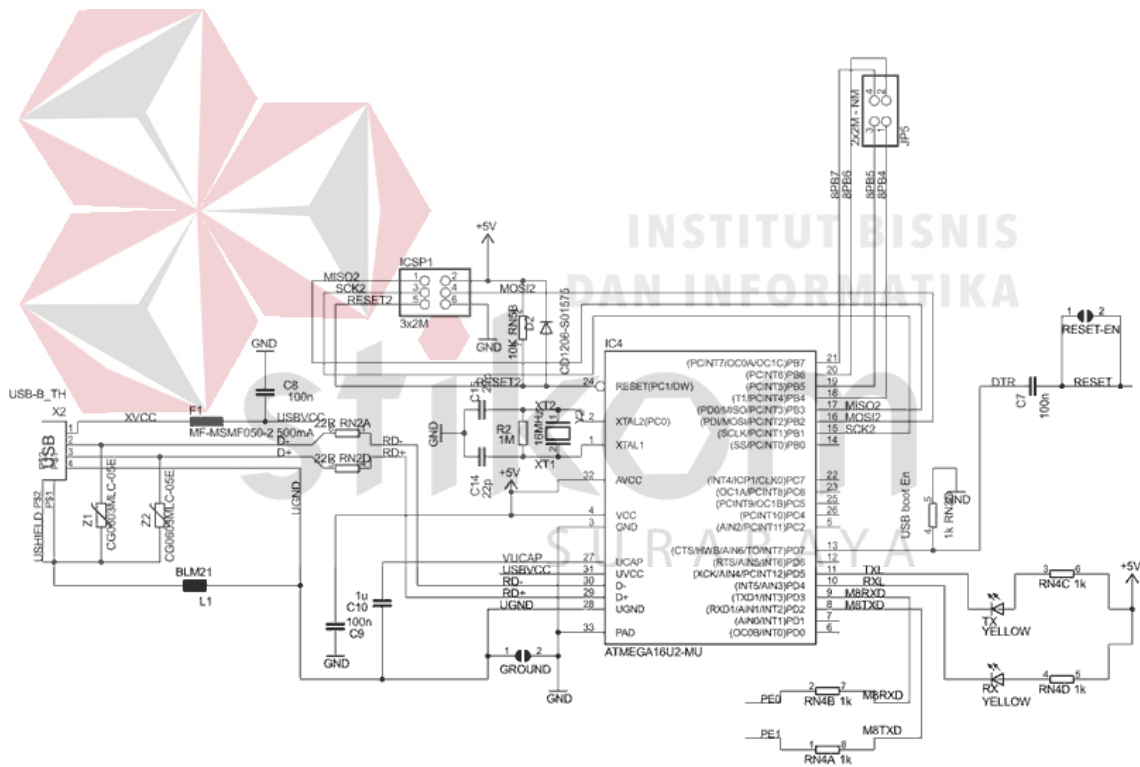
Gambar 3.5 B. Rangkaian Skematik *downloader* Arduino Mega

Skematik downloader Arduino Mega adalah sebagai USB to Serial Converter. Yaitu berfungsi sebagai penghubung antara komputer dengan mikrokontroler. Yang mana file.hex yang telah dibuat dari compile file.bas dari software Arduino IDE dimasukkan ke dalam mikrokontroler. Downloader yang umum digunakan untuk memasukkan data dari komputer ke mikrokontroler yaitu USB-ASP, namun ada pula yang tidak memakai USB-ASP yaitu menggunakan serial port untuk melakukan download programnya.



Gambar 3.5 C. Rangkaian Skematik Penurun Tegangan Arduino Mega

Pada Arduino Mega terdapat penurun tegangan karena *board* Arduino Mega beroperasi pada sebuah suplai eksternal 6 sampai 20 Volt. Jika disuplai dengan yang lebih kecil dari 7 V, kiranya pin 5 Volt mungkin mensuplai lebih kecil dari 5 Volt dan *board* Arduino Mega bisa menjadi tidak stabil. Jika menggunakan suplai yang lebih dari besar 12 Volt, voltage regulator bisa kelebihan panas dan membahayakan *board* Arduino Mega. Range yang direkomendasikan adalah 7 sampai 12 Volt.



Gambar 3.5 D. Rangkaian Skematik ATMEGA16U2 Arduino Mega

Sebuah *chip* ATmega 16U2 (ATmega 8U2 pada papan Revisi 1 dan Revisi 2) yang terdapat pada papan digunakan sebagai media komunikasi serial melalui USB dan muncul sebagai COM *Port Virtual* (pada *Device* komputer) untuk berkomunikasi dengan perangkat lunak pada komputer.

## BAB IV

### HASIL DAN PEMBAHASAN

Pada sub bab ini melakukan hasil dan pembahasn pada pengujian sistem yang telah dilakukan oleh penulis terhadap perangkat keras serta perangkat lunak dari hasil penelitian pembuatan *mobile robot* pendeteksi bola.

#### 4.1 Pengujian Arduino

Pengujian Arduino dilakukan dengan memasukan skrip program sederhana pada Arduino menggunakan aplikasi Arduino IDE. Arduino yang baik dapat mengeksekusi program dengan baik.

##### 4.1.1. Tujuan

Pengujian ini dilakukan untuk mengetahui apakah Arduino yang digunakan tidak mengalami kerusakan. Sehingga saat arduino digunakan pada sistem dapat membantu sistem berjalan dengan baik.

##### 4.1.2. Alat yang digunakan

Alat yang digunakan untuk melakukan pengujian antara lain :

- a. Kabel usb
- b. Arduino Mega 2560
- c. Komputer/laptop
- d. *Software* Arduino IDE

##### 4.1.3. Prosedur Pengujian

- a. Hubungkan Arduino dengan kabel usb
- b. Nyalakan komputer kemudian hubungkan kabel usb tadi dengan komputer.

- c. Buka *software* Arduino IDE dan isi perintah dalam bahasa C. Sebagai contoh penulis memasukkan perintah sebagai berikut :

```
void setup()
{
    Serial.begin(9600);
    Serial.println("Cek Mulai :");
}

int i=0;

void loop()
{
    Serial.print("Data ke");
    Serial.println(i);
    delay(1000);
    i++;
}
```

- d. Apabila telah selesai untuk mengisi perintah, maka tekan “*Verify*” untuk mengecek apabila terdapat perintah yang salah dalam bahasa C. Dan tekan “*Upload*” untuk memasukkan perintah tersebut ke dalam Arduino Mega 2560.
- e. Setelah *program* telah berhasil dimasukkan, maka tekan *icon Serial monitor* pada kanan atas. Maka akan muncul tampilan serial monitor.



f. Setelah window serial monitor muncul, amati kiriman data serial oleh Arduino.

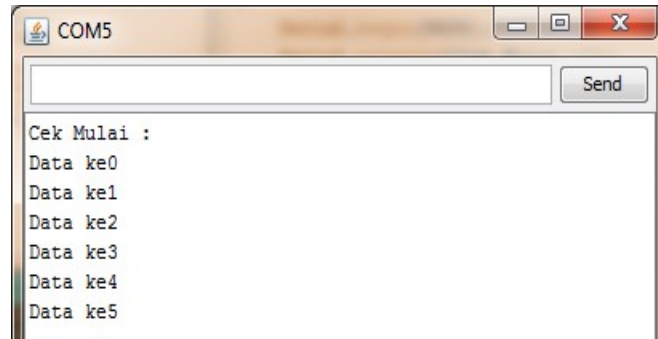
#### 4.1.4. Hasil Pengujian

Hasil dari pengujian pengisian program ke Arduino dapat dilihat pada Gambar 4.2. Lingkaran merah menunjukkan bahwa Arduino yang digunakan berhasil diisi dengan program yang telah ditulis dalam *software* Arduino IDE.



Gambar 4.1 Upload program berhasil

Program yang dimasukan kedalam Arduino merupakan program untuk mengirimkan data menggunakan serial. Proses pengiriman ini apabila Arduino masih dihubungkan dengan USB PC maka kita dapat menerima data yang dikirim menggunakan menu serial monitor pada *software* Arduino IDE. Hasil dari serial monitor dapat dilihat pada Gambar 4.1.



Gambar 4.2 *Program* berhasil berjalan

Gambar 4.2 menunjukkan bahwa data dikirimkan sesuai dengan perintah program yang telah diisi pada Arduino. Dengan begitu Arduino ini dapat bekerja dengan baik, dan dapat digunakan untuk sistem.

## 4.2 Pengujian *Driver Motor*

*Driver motor* digunakan sebagai perantara antara mikrokontroler dengan motor DC agar mikrokontroler dapat mengatur pergerakan dari motor DC dan juga kecepatan yang diinginkan. Dalam pengujian ini mikrokontroler diberikan program yang mengatur pergerakan dari motor DC.

### 4.2.1. Tujuan Pengujian *Driver Motor*

Pengujian ini bertujuan untuk melihat kinerja *Minimum System* telah dapat mengatur pergerakan dan kecepatan dari roda melalui *input* yang diberikan pada *driver motor*.

### 4.2.2. Alat yang Digunakan pada Pengujian *Driver Motor*

Untuk melakukan percobaan ini maka diperlukan beberapa alat sebagai berikut.

1. Kabel usb
2. Arduino Mega 2560

3. Komputer/laptop
4. *Software* Arduino IDE
5. *Driver motor*.
6. Motor DC.
7. *Power supply* 12V

#### **4.2.3. Prosedur Pengujian *Driver Motor***

Prosedur pengujian alat :

1. Hubungkan *power*, pin *direction* dan PWM pada motor *driver* ke *microcontroller*.
2. Hubungkan motor DC ke motor *driver*.
3. Aktifkan *power supply* 12V dan hubungkan pada *power* motor *driver*.
4. Sambungkan *Minimum System* dengan kabel USB *downloader* kemudian sambungkan pada laptop atau PC.
5. Buka program *Arduini IDE* buat program sederhana untuk menjalankan motor *verify* program yang di *upload* untuk memastikan tidak ada kesalahan *syntax* pada program. *Upload* program untuk mengatur pergerakan motor yang telah dibuat kedalam *Minimum System*
6. Setelah dipastikan tidak ada lagi *error*, maka yang dilakukan *upload* program kedalam mikrokontroler dari *Arduino IDE*, maka proses *upload* file ekstensi “.hex” kedalam mikrokontroler berhasil dan selesai.
7. Lihat pergerakan roda sesuai dengan masukan pada program.

#### 4.2.4. Hasil Pengujian *driver* Motor

Pengujian pergerakan roda dengan masukan dari mikrokontroler dan *driver motor* sebagai alat perantara dapat dilihat hasilnya pada tabel 4.1.

Tabel 4.1 Hasil pengujian pergerakan *driver* motor

No	Motor	Port	Output	Keterangan
1	Motor kanan	Dir 0	0	Motor Maju
		Dir 1	1	
		Dir 0	1	Motor Mundur
		Dir 1	0	
2	Motor kiri	Dir 0	0	Motor Maju
		Dir 1	1	
		Dir 0	1	Motor Mundur
		Dir 1	0	
3	Motor Belakang	Dir 0	0	Motor Maju
		Dir 1	1	
		Dir 0	1	Motor Mundur
		Dir 1	0	
4	Motor kanan	Dir 0	0	Motor Berhenti
		Dir 1	0	
	Motor Kiri	Dir 0	0	Motor Berhenti
		Dir 1	0	
	Motor Belakang	Dir 0	0	Motor Berhenti
		Dir 1	0	

#### 4.3 Pengujian Sinkronisasi Pembacaan Koordinat Sensor Kamera

Pembacaan koordinat yaitu hasil pembacaan sensor kamera saat mendeteksi objek. Dalam pengujian ini mikrokontroler diberikan program yang mengatur penerimaan koordinat dari sensor kamera.

#### **4.3.1. Tujuan Pengujian Sinkronisasi Pembacaan Koordinat Sensor Kamera**

Pengujian ini bertujuan untuk melihat hasil nilai dari koordinat sensor kamera dan hasil pembacaan koordinat yang dikirim ke mikrokontroler yang berada pada *mobile robot*.

#### **4.3.2. Alat Yang Digunakan Pada Pengujian Sinkronisasi Pembacaan Koordinat Sensor Kamera**

Untuk melakukan percobaan ini maka diperlukan beberapa alat sebagai berikut.

1. Kabel usb
2. Arduino Mega 2560
3. Komputer/laptop
4. *Software* Arduino IDE
5. *Driver motor*.
6. Motor DC.
7. *Power supply* 12V
8. Sensor kamera

#### **4.3.3. Prosedur Pengujian Sinkronisasi Pembacaan Koordinat Sensor Kamera**

Prosedur pengujian alat :

1. Hubungkan kabel USB dari mikrokontroler ke laptop yang berfungsi sebagai komunikasi serial.
2. Hubungkan sensor kamera ke laptop untuk menjalankan proses deteksi obyek.

3. Buka program Arduino IDE buat program sederhana untuk menerima data dari komunikasi serial.
4. Setelah dipastikan tidak ada lagi *error*, maka yang dilakukan *upload* program kedalam *microcontroller* dari Arduino IDE, maka proses *upload* file ekstensi “.hex” kedalam *microcontroller* berhasil dan selesai.
5. Jalankan proses pendeteksi obyek pada laptop
6. Lihat hasil deteksi obyek dari sensor kamera dengan pergerakan robot.

#### 4.3.4. Hasil Pengujian Sinkronisasi Pembacaan Koordinat Sensor Kamera

Pengujian sinkronisasi pembacaan koordinat sensor kamera dengan hasil pembacaan pada mikrokontroler dapat dilihat hasilnya pada tabel 4.2.

Tabel 4.2 Hasil pengujian sinkronisasi pembacaan koordinat sensor kamera.

Pembacaan Koordinat Sensor Kamera Pada Proses Mendeteksi Bola	Pembacaan Koordinat Yang Diterima Mikrokontroler Pada Mobile Robot
Koordinat X 60	Koordinat X 60
Koordinat X 100	Koordinat X 100
Koordinat X 140	Koordinat X 140
Koordinat X 180	Koordinat X 180
Koordinat X 220	Koordinat X 220
Koordinat X 260	Koordinat X 260
Koordinat X 300	Koordinat X 300
Koordinat X 400	Koordinat X 400
Koordinat X 440	Koordinat X 440
Koordinat X 480	Koordinat X 480
Koordinat X 520	Koordinat X 520
Koordinat X 560	Koordinat X 560

#### 4.4 Pengujian Sistem

Pengujian ini merupakan pengujian pergerakan robot untuk menuju ke arah bola. data yang diolah yaitu data yang berupa koordinat yang telah diambil dari sensor kamera. Analisis dilakukan dengan melakukan beberapa pengujian.

##### 4.4.1. Tujuan Pengujian Pengujian Sistem

Pengujian ini bertujuan untuk menguji robot dapat bergerak dengan baik pada saat sensor kamera telah mendeteksi keberadaan bola, dan untuk mengetahui respon sistem terhadap *set point* yang dituju, sehingga sesuai dengan keinginan dan kebutuhan pada tugas akhir ini

##### 4.4.2. Alat Yang Digunakan Pada Pengujian Sistem

1. Mobile robot
2. PC atau Laptop
3. Program IDE
4. USB Downloader
5. Arduino Mega.
6. Sensor kamera
7. Baterai 12 v
8. Roda omni
9. Motor DC
10. Kamera

##### 4.4.3. Prosedur Pengujian Sistem

1. Hubungkan semua komponen yang ada pada robot seperti *Minimum System*, Motor DC, *Driver Motor*, kamera dan power.

2. Aktifkan laptop dan jalankan program Arduino IDE.
3. Upload seluruh program yang sudah dibuat pada *Minimum System*.
4. Menentukan *Set point* atau koordinat yang diinginkan.
5. Menentukan Koordinat awal pengujian.
6. Komunikasi kamera ke Arduino dengan menggunakan komunikasi serial.
7. Menghubungkan Arduino dengan laptop atau computer menggunakan kabel USB

#### 4.4.4. Hasil Pengujian sistem

Dari prosedur pengujian sistem, yaitu untuk mengetahui respon yang dihasilkan dari pergerakan robot untuk mencapai titik stabil dengan menggunakan nilai  $K_p$ ,  $K_i$ ,  $K_d$  yang berbeda-beda.

##### A. Pengujian dengan nilai $K_p = 2,8$ , $K_i = 5$ dan $K_d = 0,8$ .

Pada pengujian ini diberikan nilai  $K_p = 2,8$ ,  $K_i = 5$  dan  $K_d = 0,8$ . Pengujian dilakukan dengan mengatur nilai koordinat X awal yang berbeda-beda terhadap *set point* koordinat  $X=330$ , dan nilai Y lebih dari 440.

Tabel 4.3. Tabel pengujian pertama dengan nilai  $K_p = 2,8$ ,  $K_i = 5$  dan  $K_d = 0,8$ .

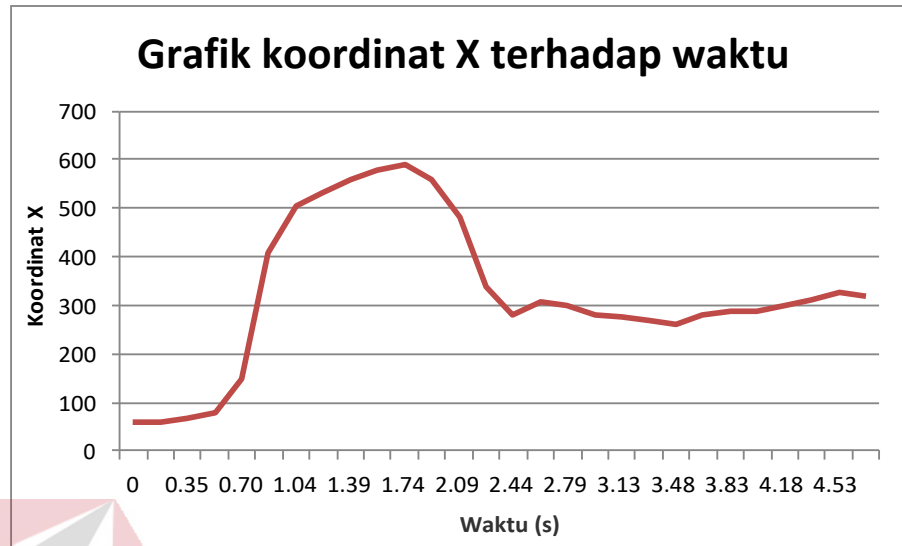
koordinat X awal = 60		
Waktu (detik)	Nilai Koordinat	
	X	Y
0	60	210
0.17	60	212
0.35	68	218
0.52	80	224
0.7	148	226
0.87	408	229
1.04	506	232
1.22	532	234
1.39	560	238



1.57	580	245
1.74	590	248
1.91	560	259
2.09	480	267
2.26	340	278
2.44	280	291
2.61	308	310
2.79	298	312
2.96	280	322
3.13	278	324
3.31	270	329
3.48	260	331
3.66	279	347
3.83	288	349
4	290	358
4.18	298	371
4.35	310	382
4.53	326	424
4.7	320	440

Tabel 4.3 menunjukkan hasil output sumbu X dari sensor kamera berdasarkan nilai keluaran koordinat pada saat robot bergerak menuju bola dengan nilai koordinat X yang diinginkan adalah 330. Dengan jarak 1,5 meter robot sampai tujuan pada detik ke 4,7 dan berada di koordinat 320, terjadi selisih atau eror dari koordinat tersebut yaitu sebesar 10 koordinat kamera.

Sedangkan hasil *plotting* dari nilai pergerakan robot menurut koordinat X ditunjukkan oleh Gambar 4.3



Gambar 4.3 Hasil *plotting* pengujian pergerakan robot menurut koordinat X.

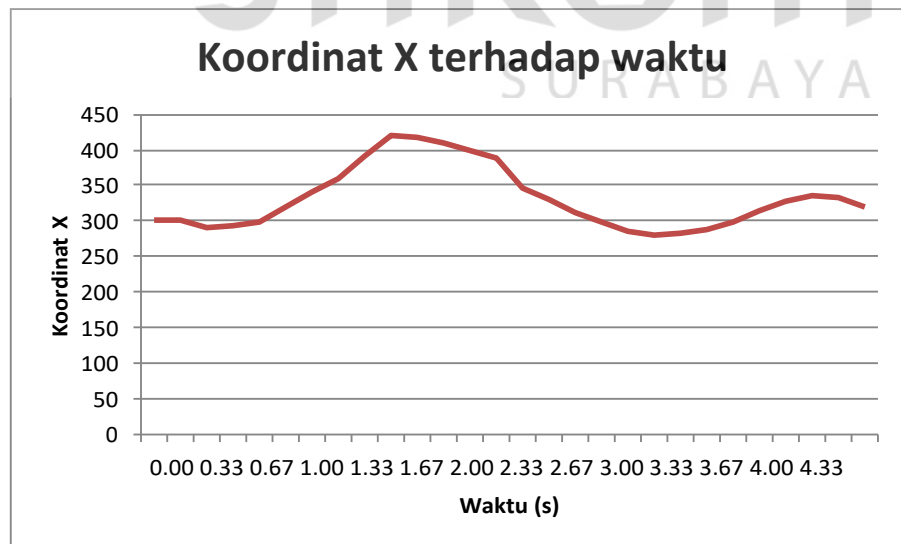
Tabel 4.4 Tabel pengujian kedua dengan nilai  $K_p = 2,8$ ,  $K_i = 5$  dan  $K_d = 0,8$ .

koordinat X awal = 330		
Waktu (detik)	Nilai Koordinat	
	X	Y
0	330	218
0.17	328	221
0.33	290	224
0.5	294	231
0.67	298	234
0.83	319	236
1	340	240
1.17	360	244
1.33	390	246
1.5	420	255
1.67	418	262
1.83	408	275
2	400	279
2.17	388	285
2.33	346	296
2.5	329	316
2.67	312	320
2.83	298	328

3	286	329
3.17	280	336
3.33	282	335
3.5	288	351
3.67	298	354
3.83	315	361
4	328	373
4.17	336	387
4.33	332	428
4.5	320	441

Tabel 4.4 menunjukkan hasil output sumbu X dari sensor kamera berdasarkan nilai keluaran koordinat pada saat robot bergerak menuju bola dengan nilai koordinat X yang diinginkan adalah 330. Dengan jarak 1,5 meter robot sampai tujuan pada detik ke 4.50 dan berada di koordinat 320 terjadi selisih atau eror dari koordinat tersebut yaitu sebesar 10 koordinat kamera.

Sedangkan hasil *plotting* dari nilai pergerakan robot menurut koordinat x ditunjukkan oleh Gambar 4.4.



Gambar 4.4 Hasil *plotting* pengujian pergerakan robot menurut koordinat X.

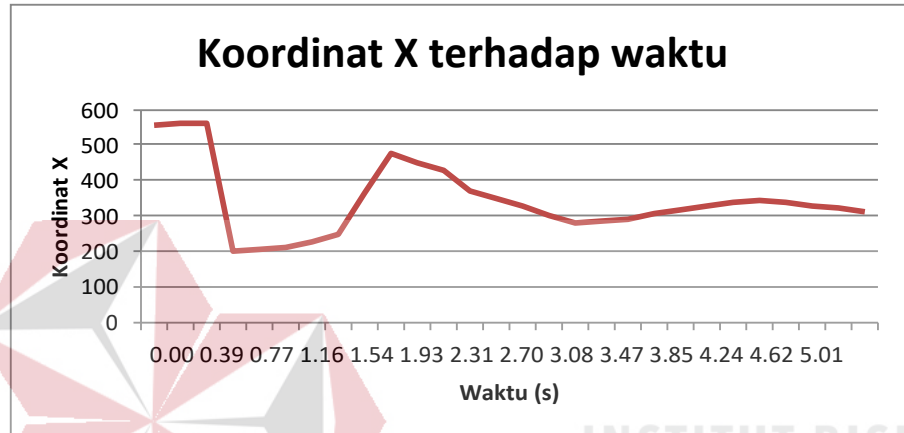
Tabel 4.5. Tabel pengujian ketiga dengan nilai  $K_p = 2,8$  ,  $K_i = 5$  dan  $K_d = 0,8$ .

koordinat X awal = 560		
Waktu (detik)	Nilai Koordinat	
	X	Y
0	558	209
0.19	560	211
0.39	562	210
0.58	200	214
0.77	208	215
0.96	212	216
1.16	227	218
1.35	247	220
1.54	367	226
1.73	474	235
1.93	448	237
2.12	429	246
2.31	370	255
2.5	350	267
2.7	328	278
2.89	300	290
3.08	278	299
3.27	285	312
3.47	290	325
3.66	308	338
3.85	318	356
4.04	327	365
4.24	340	356
4.43	342	367
4.62	338	374
4.81	330	408
5.01	321	425
5.2	314	443

Tabel 4.5 menunjukkan hasil output sumbu X dari sensor kamera berdasarkan nilai keluaran koordinat pada saat robot bergerak menuju bola dengan nilai koordinat X yang diinginkan adalah 330. Dengan jarak 1,5 meter robot

sampai tujuan pada detik ke 5,2 dan berada di koordinat 314 terjadi selisih atau eror dari koordinat tersebut yaitu sebesar 16 koordinat kamera.

Sedangkan hasil *plotting* dari nilai pergerakan robot menurut koordinat x ditunjukkan oleh Gambar 4.5.



Gambar 4.5 Hasil *plotting* pengujian pergerakan robot menurut koordinat X.

Tabel 4.6. Hasil respon dari pengujian pergerakan robot dengan nilai  $K_p = 2.8$ ,  $K_i = 5$ ,  $K_d = 0.8$ .

Set point Koordinat X 330	Hasil pengujian pergerakan robot			
	Koordinat X awal	Rise time (s)	Overshoot (%)	Settling time (s)
60.00		0.70	84.38	2.19
330.00		0.80	31.25	3.2
560.00		0.47	36.31	1.47

Keterangan :

1. *Rise Time* (Waktu naik) adalah ukuran waktu yang di ukur mulai dari respon  $t=0$  sampai dengan respon memotong sumbu *steady state* yang pertama.
2. *Overshoot* adalah Nilai relatif yang menyatakan perbandingan antara nilai maksimum respon (*overshoot*) yang melampaui nilai *steady state* dibanding dengan nilai *steady state*.

3. *Settling time* (Waktu tunak) adalah ukuran waktu yang menyatakan respon

telah masuk  $\pm 5\%$ , atau  $\pm 2\%$ , atau  $\pm 0.5\%$  dari keadaan *steady state*.

*Set point* koordinat X sebesar 330 yaitu kondisi pada saat robot berada hampir di posisi lurus dengan target atau dengan bola, nilai koordinat yang lebih kecil dari 330 yaitu posisi bola berada disamping kiri dari jangkauan kamera, nilai koordinat yang lebih besar dari 330 yaitu posisi bola berada disamping kanan dari jangkauan kamera.

*Set point* koordinat Y sebesar 440 yaitu kondisi dimana batas kamera dapat mendeteksi bola pada jarak yang paling dekat atau sekitar 15cm dari depan kamera, koordinat inilah yang digunakan sebagai *set point* dalam melakukan pemberhentian atau sebagai indikator bahwa bola sudah berada di depan robot.

#### **B. Pengujian dengan nilai $K_p = 0.8$ , $K_i = 1$ , $K_d = 0.08$ .**

Pada pengujian ini diberikan nilai  $K_p = 0.8$ ,  $K_i = 1$ ,  $K_d = 0.08$ . Pengujian dilakukan dengan mengatur nilai koordinat X awal yang berbeda-beda terhadap *set point* koordinat X=330, dan nilai Y lebih dari 440.

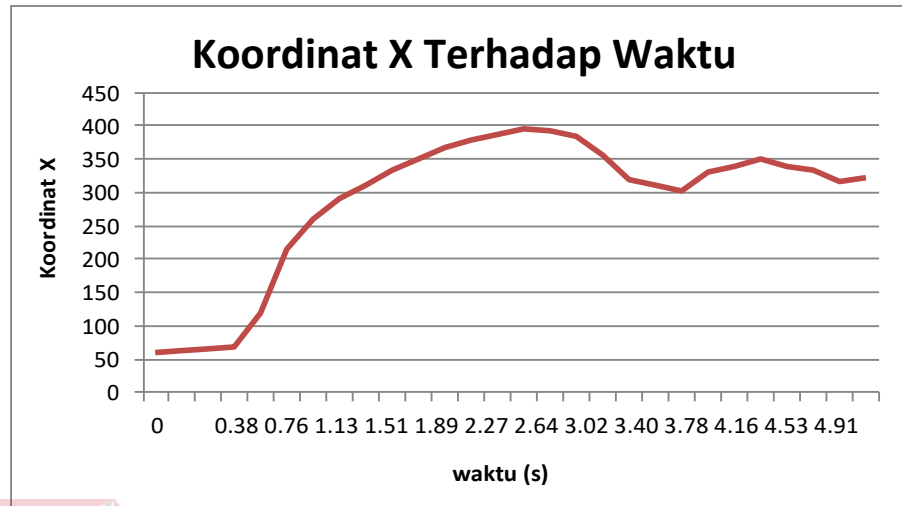
Tabel 4.7. Tabel pengujian pertama dengan nilai  $K_p = 0.8$ ,  $K_i = 1$ ,  $K_d = 0.08$ .

Koordinat X awal = 60		
Waktu (detik)	Nilai Koordinat	
	X	Y
0	60	206
0.19	64	210
0.38	66	217
0.57	68	224
0.76	120	227
0.94	216	231
1.13	260	233
1.32	292	234
1.51	312	237

1.7	332	246
1.89	350	251
2.08	366	263
2.27	378	272
2.46	388	281
2.64	394	295
2.83	392	310
3.02	384	316
3.21	356	327
3.4	320	328
3.59	310	332
3.78	302	333
3.97	330	348
4.16	340	352
4.34	350	362
4.53	340	373
4.72	332	383
4.91	317	425
5.1	321	446

Tabel 4.7 menunjukkan hasil output sumbu X dari sensor kamera berdasarkan nilai keluaran koordinat pada saat robot bergerak menuju bola dengan nilai koordinat X yang diinginkan adalah 330. Dengan jarak 1,5 meter robot sampai tujuan pada detik ke 5,1 dan berada di koordinat 321, terjadi selisih atau eror dari koordinat tersebut yaitu sebesar 11 koordinat kamera.

Sedangkan hasil *plotting* dari nilai pergerakan robot menurut koordinat X ditunjukkan oleh Gambar 4.6.



Gambar 4.6 Hasil *plotting* pengujian pergerakan robot menurut koordinat X.

Tabel 4.8. Tabel pengujian kedua dengan nilai  $K_p = 0.8$ ,  $K_i = 1$ ,  $K_d = 0.08$ .

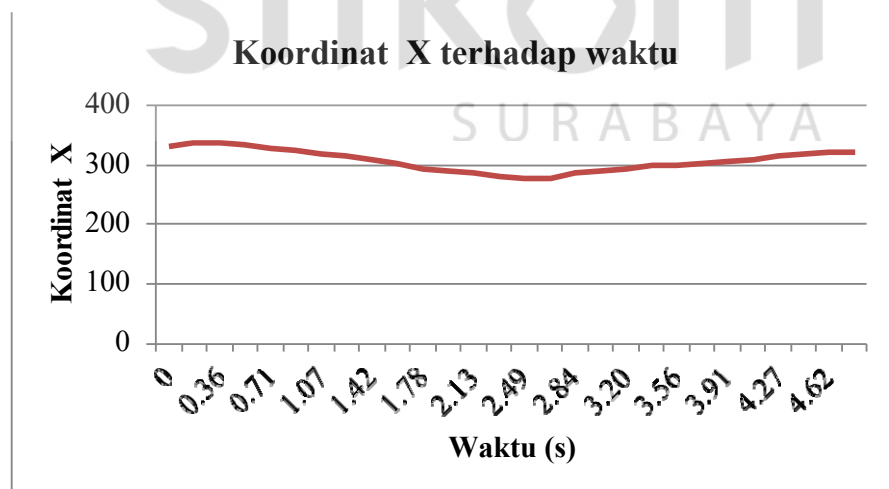
Koordinat X awal = 330		
Waktu (detik)	Nilai Koordinat	
	X	Y
0	330	212
0.18	336	214
0.36	338	221
0.53	334	225
0.71	328	231
0.89	325	236
1.07	318	240
1.24	316	244
1.42	308	248
1.6	302	251
1.78	292	259
1.96	290	265
2.13	288	272
2.31	280	275
2.49	276	277
2.67	278	308
2.84	286	309
3.02	290	323
3.2	294	329



3.38	298	338
3.56	300	352
3.73	302	371
3.91	304	377
4.09	310	387
4.27	314	408
4.44	318	418
4.62	320	433
4.8	322	445

Tabel 4.8 menunjukkan hasil output sumbu X dari sensor kamera berdasarkan nilai keluaran koordinat pada saat robot bergerak menuju bola dengan nilai koordinat X yang diinginkan adalah 330. Dengan jarak 1,5 meter robot sampai tujuan pada detik ke 4.80 dan berada di koordinat 322 terjadi selisih atau eror dari koordinat tersebut yaitu sebesar 12 koordinat kamera.

Sedangkan hasil *plotting* dari nilai pergerakan robot menurut koordinat x ditunjukkan oleh Gambar 4.7



Gambar 4.7 Hasil *plotting* pengujian pergerakan robot menurut koordinat X.

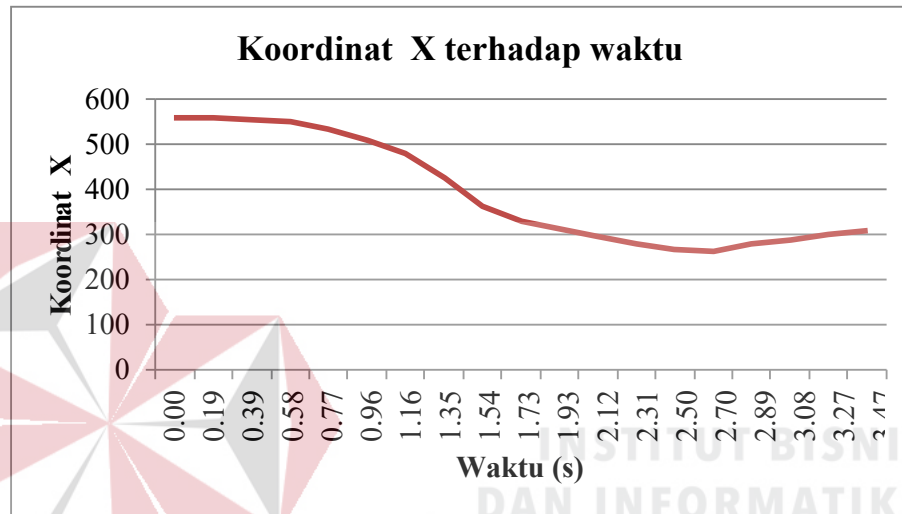
Tabel 4.9. Tabel pengujian ketiga dengan nilai  $K_p = 0.8$ ,  $K_i = 1$ ,  $K_d = 0.08$ .

Koordinat X awal = 560		
Waktu (detik)	Nilai Koordinat	
	X	Y
0	560	214
0.19	558	218
0.39	554	220
0.58	550	224
0.77	534	226
0.96	508	229
1.16	480	236
1.35	426	238
1.54	364	242
1.73	330	245
1.93	312	253
2.12	298	261
2.31	280	276
2.5	268	294
2.7	263	306
2.89	280	320
3.08	290	334
3.27	302	340
3.47	308	346
3.66	314	348
3.85	320	356
4.04	326	367
4.24	335	388
4.43	340	392
4.62	344	412
4.81	346	426
5.01	349	437
5.2	352	442

Tabel 4.9 menunjukkan hasil output sumbu X dari sensor kamera berdasarkan nilai keluaran koordinat pada saat robot bergerak menuju bola dengan nilai koordinat X yang diinginkan adalah 330. Dengan jarak 1,5 meter robot

sampai tujuan pada detik ke 5,2 dan berada di koordinat 352 terjadi selisih atau eror dari koordinat tersebut yaitu sebesar 22 koordinat kamera..

Sedangkan hasil *plotting* dari nilai pergerakan robot menurut koordinat x ditunjukkan oleh Gambar 4.8.



Gambar 4.8 Hasil *plotting* pengujian pergerakan robot menurut koordinat X.

Tabel 4.10. Hasil respon dari pengujian pergerakan robot dengan nilai  $K_p = 0.8$ ,

$K_i = 1$ ,  $K_d = 0.08$ .

Set point Koordinat X 330	Hasil pengujian pergerakan robot			
	Koordinat X awal	Rise time (s)	Overshoot (%)	Settling time (s)
60		1.32	22.74	4.76
330		0.56	14.28	1.80
560		1.34	25.28	4.85

Keterangan :

1. *Rise Time* (Waktu naik) adalah ukuran waktu yang di ukur mulai dari respon  $t=0$  sampai dengan respon memotong sumbu *steady state* yang pertama.
2. *Overshoot* adalah Nilai relatif yang menyatakan perbandingan antara nilai maksimum respon (*overshoot*) yang melampaui nilai *steady state* dibanding dengan nilai *steady state*.
3. *Settling time* (Waktu tunak) adalah ukuran waktu yang menyatakan respon telah masuk  $\pm 5\%$ , atau  $\pm 2\%$ , atau  $\pm 0.5\%$  dari keadaan *steady state*.

*Set point* koordinat X sebesar 330 yaitu kondisi pada saat robot berada hampir di posisi lurus dengan target atau dengan bola, nilai koordinat yang lebih kecil dari 330 yaitu posisi bola berada disamping kiri dari jangkauan kamera, nilai koordinat yang lebih besar dari 330 yaitu posisi bola berada disamping kanan dari jangkauan kamera.

*Set point* koordinat Y sebesar 440 yaitu kondisi dimana batas kamera dapat mendeteksi bola pada jarak yang paling dekat atau sekitar 15cm dari depan kamera, koordinat inilah yang digunakan sebagai *set point* dalam melakukan pemberhentian atau sebagai indikator bahwa bola sudah berada di depan robot.

### C. Pengujian dengan nilai $K_p = 5.8$ , $K_i = 9$ , $K_d = 4.9$ .

Pada pengujian ini diberikan nilai  $K_p = 5.8$ ,  $K_i = 9$ ,  $K_d = 4.9$ . Pengujian dilakukan dengan mengatur nilai koordinat X awal yang berbeda-beda terhadap *set point* koordinat  $X=330$ , dan nilai Y lebih dari 440.

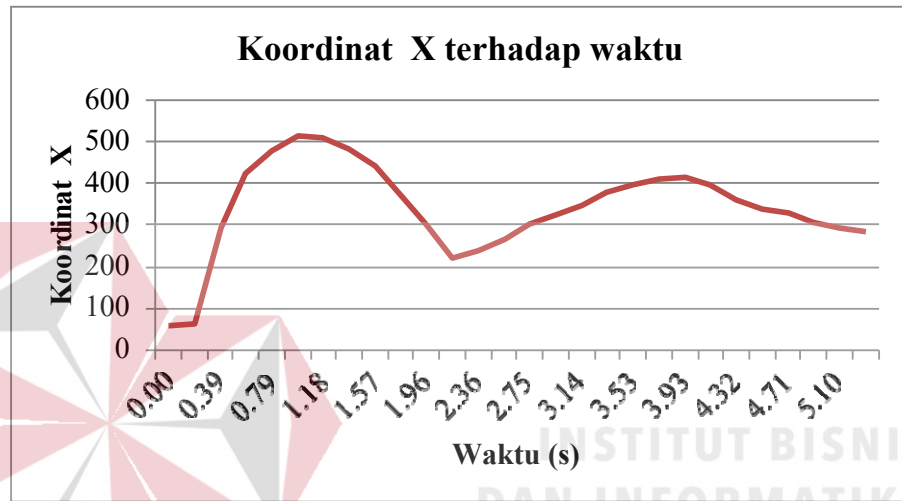
Tabel 4.11. Tabel pengujian pertama dengan nilai  $K_p = 5.8$ ,  $K_i = 9$ ,  $K_d = 4.9$ .

Koordinat X awal = 60		
Waktu (detik)	Nilai Koordinat	
	X	Y
0	60	218
0.2	64	219
0.39	294	225
0.59	424	226
0.79	480	234
0.98	515	239
1.18	508	241
1.37	484	247
1.57	442	250
1.77	370	256
1.96	302	263
2.16	220	266
2.36	237	270
2.55	268	274
2.75	300	279
2.94	325	306
3.14	348	311
3.34	379	322
3.53	398	325
3.73	412	334
3.93	416	348
4.12	398	370
4.32	362	375
4.51	340	381
4.71	327	402
4.91	306	411
5.1	292	427
5.3	286	442

Tabel 4.11 menunjukkan hasil output sumbu X dari sensor kamera berdasarkan nilai keluaran koordinat pada saat robot bergerak menuju bola dengan nilai koordinat X yang diinginkan adalah 330. Dengan jarak 1,5 meter robot

sampai tujuan pada detik ke 5,3 dan berada di koordinat 286, terjadi selisih atau eror dari koordinat tersebut yaitu sebesar 44 koordinat kamera.

Sedangkan hasil *plotting* dari nilai pergerakan robot menurut koordinat X ditunjukkan oleh Gambar 4.9.



Gambar 4.9 Hasil *plotting* pengujian pergerakan robot menurut koordinat X.

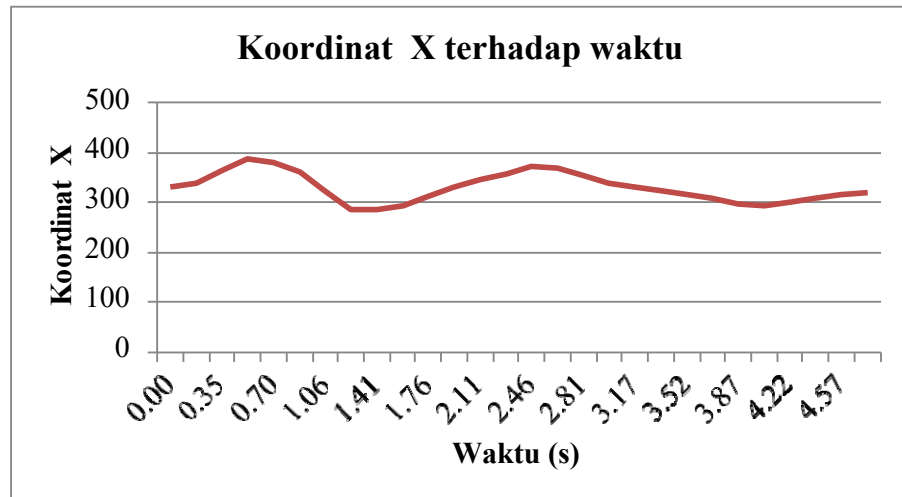
Tabel 4.12. Tabel pengujian kedua dengan nilai  $K_p = 5.8$ ,  $K_i = 9$ ,  $K_d = 4.9$ .

Koordinat X awal = 330		
Waktu (detik)	Nilai Koordinat	
	X	Y
0	330	200
0.18	340	206
0.35	364	219
0.53	388	227
0.7	380	230
0.88	360	236
1.06	324	239
1.23	284	241
1.41	284	243
1.58	292	248
1.76	312	252
1.94	329	265

2.11	345	272
2.29	356	282
2.46	372	297
2.64	368	313
2.81	352	318
2.99	340	325
3.17	332	326
3.34	324	335
3.52	316	338
3.69	307	355
3.87	298	364
4.05	294	373
4.22	300	387
4.4	308	400
4.57	315	426
4.75	320	440

Tabel 4.12 menunjukkan hasil output sumbu X dari sensor kamera berdasarkan nilai keluaran koordinat pada saat robot bergerak menuju bola dengan nilai koordinat X yang diinginkan adalah 330. Dengan jarak 1,5 meter robot sampai tujuan pada detik ke 4.75 dan berada di koordinat 320 terjadi selisih atau eror dari koordinat tersebut yaitu sebesar 10 koordinat kamera.

Sedangkan hasil *plotting* dari nilai pergerakan robot menurut koordinat x ditunjukkan oleh Gambar 4.10.



Gambar 4.10 Hasil *plotting* pengujian pergerakan robot menurut koordinat X.

Tabel 4.13. Tabel pengujian ketiga dengan nilai  $K_p = 5.8$ ,  $K_i = 9$ ,  $K_d = 4.9$ .

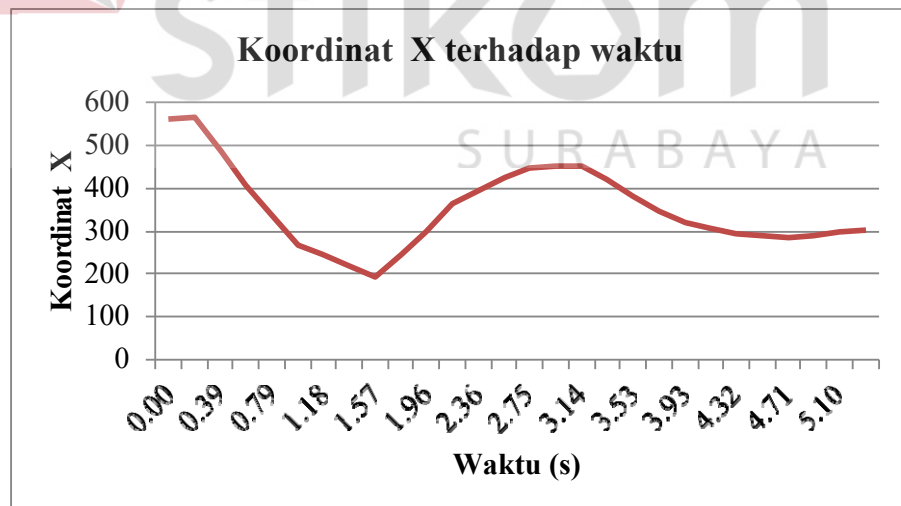
Koordinat X awal = 560		
Waktu (detik)	Nilai Koordinat	
	X	Y
0	560	209
0.2	564	211
0.39	486	210
0.59	408	214
0.79	336	215
0.98	268	216
1.18	246	218
1.37	220	220
1.57	194	226
1.77	246	235
1.96	300	237
2.16	363	246
2.36	396	255
2.55	424	267
2.75	446	278
2.94	452	290
3.14	450	299
3.34	420	312
3.53	380	325



3.73	348	338
3.93	320	356
4.12	308	365
4.32	295	356
4.51	289	367
4.71	286	374
4.91	290	408
5.1	298	425
5.3	304	443

Tabel 4.13 menunjukkan hasil output sumbu X dari sensor kamera berdasarkan nilai keluaran koordinat pada saat robot bergerak menuju bola dengan nilai koordinat X yang diinginkan adalah 330. Dengan jarak 1,5 meter robot sampai tujuan pada detik ke 5,3 dan berada di koordinat 304 terjadi selisih atau eror dari koordinat tersebut yaitu sebesar 26 koordinat kamera..

Sedangkan hasil *plotting* dari nilai pergerakan robot menurut koordinat x ditunjukkan oleh Gambar 4.11.



Gambar 4.11 Hasil *plotting* pengujian pergerakan robot menurut koordinat X.

Tabel 4.14. Hasil respon dari pengujian pergerakan robot dengan nilai  $K_p = 5.8$ ,

$$K_i = 9, K_d = 4.9.$$

<i>Set point</i> Koordinat X 330	Hasil pengujian pergerakan robot		
Koordinat X awal	Rise time (s)	Overshoot (%)	Settling time (s)
60	0.47	80.1	1.49
330	0.84	18.75	2.66
560	0.78	36.18	2.47

Keterangan :

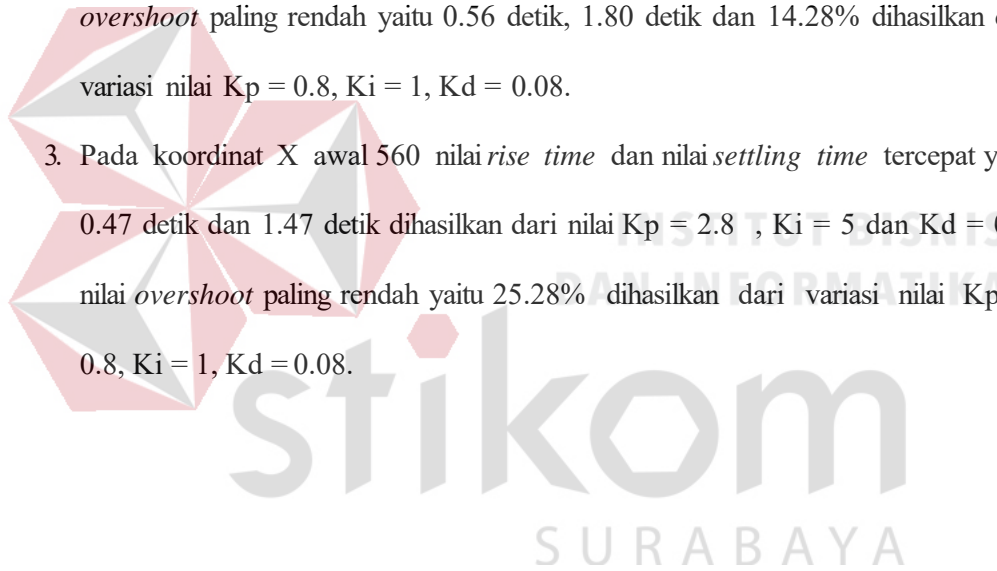
1. *Rise Time* (Waktu naik) adalah ukuran waktu yang di ukur mulai dari respon  $t=0$  sampai dengan respon memotong sumbu *steady state* yang pertama.
2. *Overshoot* adalah Nilai relatif yang menyatakan perbandingan antara nilai maksimum respon (*overshoot*) yang melampaui nilai *steady state* dibanding dengan nilai *steady state*.
3. *Settling time* (Waktu tunak) adalah ukuran waktu yang menyatakan respon telah masuk  $\pm 5\%$ , atau  $\pm 2\%$ , atau  $\pm 0.5\%$  dari keadaan *steady state*.

*Set point* koordinat X sebesar 330 yaitu kondisi pada saat robot berada hampir di posisi lurus dengan target atau dengan bola, nilai koordinat yang lebih kecil dari 330 yaitu posisi bola berada disamping kiri dari jangkauan kamera, nilai koordinat yang lebih besar dari 330 yaitu posisi bola berada disamping kanan dari jangkauan kamera.

*Set point* koordinat Y sebesar 440 yaitu kondisi dimana batas kamera dapat mendeteksi bola pada jarak yang paling dekat atau sekitar 15cm dari depan kamera, koordinat inilah yang digunakan sebagai *set point* dalam melakukan pemberhentian atau sebagai indikator bahwa bola sudah berada di depan robot.

Hasil dari berbagai variasi nilai  $K_p$ ,  $K_i$ ,  $K_d$  memberikan hasil dan respon yang berbeda pada tiga variasi yaitu:

1. Pada koordinat X awal 60 nilai *rise time* dan nilai *settling time* tercepat yaitu 4.7 detik dan 1.49 detik yang dihasilkan dari nilai  $K_p = 5.8$ ,  $K_i = 9$ ,  $K_d = 4.9$ , nilai *overshoot* paling rendah yaitu 22.74% dihasilkan dari variasi nilai  $K_p = 0.8$ ,  $K_i = 1$ ,  $K_d = 0.08$
2. Pada koordinat X awal 330 nilai *rise time*, *settling time* tercepat dan nilai *overshoot* paling rendah yaitu 0.56 detik, 1.80 detik dan 14.28% dihasilkan dari variasi nilai  $K_p = 0.8$ ,  $K_i = 1$ ,  $K_d = 0.08$ .
3. Pada koordinat X awal 560 nilai *rise time* dan nilai *settling time* tercepat yaitu 0.47 detik dan 1.47 detik dihasilkan dari nilai  $K_p = 2.8$ ,  $K_i = 5$  dan  $K_d = 0.8$ , nilai *overshoot* paling rendah yaitu 25.28% dihasilkan dari variasi nilai  $K_p = 0.8$ ,  $K_i = 1$ ,  $K_d = 0.08$ .



## BAB V

### PENUTUP

Berdasarkan Pengujian pada perangkat keras dan perangkat lunak yang digunakan dalam tugas akhir ini, maka dapat diambil kesimpulan dan saran-saran dari hasil yang diperoleh.

#### 5.1. Kesimpulan

1. Pada pengujian sinkronisasi antara pembacaan koordinat pada sensor kamera dengan hasil pembacaan koordinat yang diterima mikrokontroler pada *mobile robot* yaitu pembacaan koordinat yang diterima mikrokontroler pada *mobile robot* sesuai dengan koordinat pembacaan dari sensor kamera.
2. Pada pengujian sistem keseluruhan didapatkan hasil dari respon pergerakan robot dengan menggunakan berbagai variasi nilai  $K_p$ ,  $K_i$ ,  $K_d$ . Dari berbagai nilai variasi nilai  $K_p$ ,  $K_i$ ,  $K_d$  didapatkan nilai respon yaitu:
  - a. Pada koordinat X awal 60 nilai *rise time* dan nilai *settling time* tercepat yaitu 4.7 detik dan 1.49 detik yang dihasilkan dari nilai  $K_p = 5.8$ ,  $K_i = 9$ ,  $K_d = 4.9$ , nilai *overshoot* paling rendah yaitu 22.74% dihasilkan dari variasi nilai  $K_p = 0.8$ ,  $K_i = 1$ ,  $K_d = 0.08$
  - b. Pada koordinat X awal 330 nilai *rise time*, *settling time* tercepat dan nilai *overshoot* paling rendah yaitu 0.56 detik, 1.80 detik dan 14.28% dihasilkan dari variasi nilai  $K_p = 0.8$ ,  $K_i = 1$ ,  $K_d = 0.08$ .
  - c. Pada koordinat X awal 560 nilai *rise time* dan nilai *settling time* tercepat yaitu 0.47 detik dan 1.47 detik dihasilkan dari nilai  $K_p = 2.8$ ,  $K_i = 5$  dan

$K_d = 0.8$ , nilai *overshoot* paling rendah yaitu 25.28% dihasilkan dari variasi nilai  $K_p = 0.8$ ,  $K_i = 1$ ,  $K_d = 0.08$ .

Berdasarkan hasil pengujian diatas, pada Tugas Akhir ini pergerakan *mobile robot* yang diinginkan yaitu robot dapat dengan cepat merespon posisi keberadaan bola dan juga dapat dengan cepat bergerak menuju bola tersebut, maka dari itu pada kendali PID ini menggunakan nilai  $K_p = 2.8$ ,  $K_i = 5$  dan  $K_d = 0.8$ . Karena dengan menggunakan nilai  $K_p$ ,  $K_i$ ,  $K_d$  tersebut didapatkan respon pergerakan robot yang cepat dan nilai *overshoot* tidak terlalu tinggi, tetapi nilai *steady state* yang lama.

## 5.2. Saran

Pengembangan lebih lanjut dari penelitian Tugas Akhir ini, maka penulis memberikan beberapa saran sebagai berikut :

1. Mengganti sensor kamera dengan yang lebih baik.
2. Menggunakan *driver motor* dengan tipe atau seri yang sama dan memiliki kekuatan arus yang besar.
3. Menggunakan mikrokontroler dengan yang lebih baik supaya prosesnya lebih cepat.

## DAFTAR PUSTAKA

Arduino. (2016). *Arduino mega & Genuino mega*. Retrieved august 12, 2016, from Arduino: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560S>

Djuandi, F. (2011). *Pengenalan Arduino*. Jakarta: 2011.

Elektronika, T. (2012, July 4). *Teori Motor DC dan Jenis-jenis Motor DC*. Retrieved August 26, 2016, from Elektronika Dasar: <http://elektronika-dasar.web.id/teori-motor-dc-dan-jenis-jenis-motor-dc/>

Irfan, M. (2016). Desain Dan Implementasi Kendali Kecepatan Motor Pada Robot Dengan Empat Roda Omni Menggunakan Metode Logika Fuzzy. *Jurnal Mekanikal*, iv.

Mas Pratama, I. P., Suwedan, I. N., & Swamardika, I. B. (2013). Sistem Kontrol Pergerakan Pada Robot Line Follower Berbasis Hybrid PID-Fuzzy Logic. *PROSIDING CSGTEIS 2013*, 160-161.

Mas'ud, M. I., Hafidz, M., Harianto, & Wibowo, M. C. (2012). Kendali PID dan Logika Fuzzy Untuk Optimalisasi. *Journal of Control and Network Systems*, 2-3.

Naziq, A. (n.d.). *Pengertian Mikrokontroller*. Retrieved September 16, 2016, from Informasi Anyar: <https://sites.google.com/site/informasiterbarusekali/pengertian-mikrokontroller>

Rochmanto, R. A. (2014). Implementasi Robot Three Omni-Directional Menggunakan Kontroler Pid Pada Robot Kontes Robot Abu Indonesia (KRAI). 1-2.

Syam, R., & Erlangga, W. (2012). Rancang Bangun Omni Wheels Robot Dengan Roda Penggerak Independent. *Jurnal Mekanikal*, 215-216.

