

**PERANCANGAN ROBOT PENYEIMBANG MENGGUNAKAN  
SENSOR JARAK BERBASIS ANDROID**

**SKRIPSI**

Oleh :

**MUHAMMAD NAZRI  
NIM. 1110000271**



**PROGRAM TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS POTENSI UTAMA  
MEDAN  
2016**

**LEMBAR PENGESAHAN**

**PERANCANGAN ROBOT PENYEIMBANG MENGGUNAKAN  
SENSOR JARAK BERBASIS ANDROID**

**SKRIPSI**

Diajukan untuk Melengkapi Persyaratan Guna  
Mendapatkan Gelar Sarjana Strata Satu  
Program Studi Teknik Informatika

Oleh :

**MUHAMMAD NAZRI**  
NIM. 1110000271

Disetujui Oleh :

**Pembimbing I**



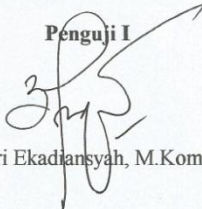
(Edy Victor Haryanto, M.Kom)

**Pembimbing II**



(Iwan Fitrianto Rahmad, M.Kom)

**Penguji I**



(Evri Ekadiansyah, M.Kom)

**Penguji II**



(Amirhud Dalimunthe, ST, M.Kom)

Medan, 22 Oktober 2016  
Diketahui dan Disahkan Oleh :

**Dekan  
Fakultas Teknik dan Ilmu Komputer**



(Ratih Puspasari, M.Kom)

**Ketua Program Studi**



(Budhi Prandi, M.Kom)

No. Dokumen : F-FTIK-21-16 Tanggal Efektif : 13 Juli 2015

No.Revisi.: 01

Halaman : 1 dari 1

Dokumen ini milik Universitas Potensi Utama, Dilarang memperbanyak atau menggunakan informasi didalamnya tanpa persetujuan Universitas Potensi Utama

**LEMBAR PERSETUJUAN SIDANG SKRIPSI**

**PERANCANGAN ROBOT PENYEIMBANG MENGGUNAKAN  
SENSOR JARAK BERBASIS ANDROID**

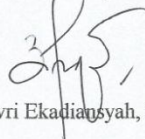
**Yang Dipersiapkan Dan Disusun Oleh :**

**MUHAMMAD NAZRI  
NIM. 1110000271**

**Telah Memenuhi Persyaratan Untuk Dipertahankan  
Didepan Dewan Penguji Pada Ujian Sidang Skripsi**

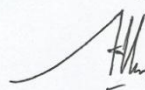
Disetujui Oleh :

Pembanding I



( Evri Ekadiansyah, M.Kom)

Pembanding II



( Amirhud Dalimunthe, ST, M.Kom)

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS POTENSI UTAMA  
MEDAN  
2016**

*No. Dokumen : F-FTIK-21-04 Tanggal Efektif : 13 Juli 2015*

*No.Revisi.: 01*

*Halaman : 1 dari 1*

*Dokumen ini milik Universitas Potensi Utama, Dilarang memperbanyak atau menggunakan informasi didalamnya tanpa persetujuan  
Universitas Potensi Utama*



DOKUMEN LEVEL  
FORM

NO. DOKUMEN  
F-FTIK-12-14

JUDUL  
JADWAL BIMBINGAN SKRIPSI

Tanggal Terbit : 07 Nov 2014

Tanggal Efektif : 14 Nov 2014

AREA  
PROGRAM STUDI

Halaman : 1 dari 1

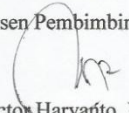
NO.REVISI  
00


### JADWAL BIMBINGAN SKRIPSI

Nama Mahasiswa : Muhammad Nazri  
NIM : 1110000271  
Program Studi : Teknik Informatika  
Judul : Perancangan Robot Penyeimbang Menggunakan Sensor Jarak Berbasis Android

NO	TANGGAL	MATERI BIMBINGAN	T. TANGAN PEMBIMBING
1	5-4-2016	Revisi Proposal	Edy
2	8-4-2016	Revisi & layout Bab I	Edy
3	20-7-2016	Az Proposal	Edy
4		Revisi Bab I	Edy
5	11-8-2016	Az Bab I	Edy
6	16-8-2016	Revisi Bab II	Edy
7	31-8-2016	Az Bab II	Edy
8	14-9-2016	Revisi	Edy
9	16-9-2016	Az Bab III	Edy
10	17-9-2016	Revisi Bab IV & V	Edy
11	28-9-2016	Az Bab IV & V	Edy
12	28-9-2016	Az Kesimpulan	Edy
13			
14			
15			

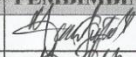

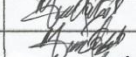

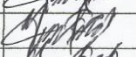
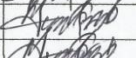
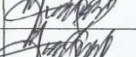
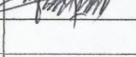
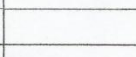
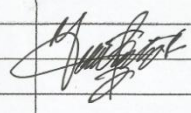
  
Ketua Program Studi  
TEKNIK INFORMATIKA  
(Budi Triandri, M.Kom)

  
Dosen Pembimbing I  
(Edy Victor Haryanto, M.Kom)

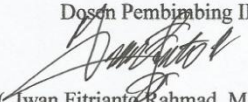
	<b>DOKUMEN LEVEL FORM</b>	<b>NO. DOKUMEN</b> F-FTIK-12-14
	<b>JUDUL</b> JADWAL BIMBINGAN SKRIPSI	Tanggal Terbit : 07 Nov 2014 Tanggal Efektif : 14 Nov 2014
<b>AREA</b> PROGRAM STUDI	Halaman : 1 dari 1	<b>NO.REVISI</b> 00

**JADWAL BIMBINGAN SKRIPSI**

Nama Mahasiswa : Muhammad Nazri  
 NIM : 1110000271  
 Program Studi : Teknik Informatika  
 Judul : Perancangan Robot Penyeimbang Menggunakan Sensor Jarak Berbasis Android

NO	TANGGAL	MATERI BIMBINGAN	T. TANGAN PEMBIMBING
1	9-11-2016	Proposal - Revisi	
2	10-6-2016	Proposal - OK	
3	13-9-2016	Bab I - OK	
4	11-8-2016	Bab II - OK	
5	1-9-2016	Bab III - Revisi	
6	2-9-2016	Bab III - OK	
7	13-9-2016	Bab IV - Revisi	
8	14-9-2016	Bab IV - OK	
9	17-9-2016	Bab V - OK	
10			
11			
12	9/16	Lampiran Lembar Kerja	
13	7/10	KSC Keseluruhan	
14			
15			

  
 Ketua Program Studi  
 (Budi Triandi, M.Kom)

Dosen Pembimbing II  
  
 (Iwan Fitrianto Rahmad, M.Kom)

Untuk ayah dan ibu tersayang

Satu malam satu lembar saja...!!

Diam & mulailah belajar...!!

Bukankah janjimu ingin jadi SARJANA?

Janganlah membuat mereka meneteskan air mata!

Baju toga itu, mengeringkan semua keringat mereka!

Menghapus air mata mereka!

Membayar semua pengorbanan mereka!

Ingat...! bukan emas & permata sebagai bentuk balas jasa!

Hanya kata – kata sederhana!

SARJANA...Saja!!

Lupakah kau waktu mereka mengantarmu pertama kali masuk kuliah?

Mereka pulang lalu bercerita kepada siapa saja bahwa anak mereka sekarang

kuliah dan menjadi calon SARJANA!

Mereka lalu menjual apa pun yang ada!

Mereka mulai berhemat uang belanja!

Tetap bekerja walau HUJAN DAN PANAS! Yang mereka rasakan!

Mencoba tetap tersenyum walau hidup dalam kekurangan, kita tak pernah tau mereka berlari kesana kemari mencari pinjaman saat kita tiba – tiba atau meminta sesuatu untuk dibayar.

Semua itu demi ANAKNYA yang tercinta,

(DEDIKASI UNTUK AYAH DAN IBU TERCINTA)

## **Lembar Pengakuan**

“Saya akui ini adalah hasil kerja saya sendiri kecuali kutipan dan ringkasan yang masing – masing telah saya jelaskan sumbernya”.

Tanda Tangan :

Nim : 1110000271

Nama Penulis : Muhammad Nazri

Tanggal : 17 September 2016

## **ABSTRAK**

*Animatronik berasal dari kata animation dan electronic. Ciri utamanya adalah menggunakan robot, bebau elektronik dan menggunakan semacam boneka mekanis. Seperti boneka yang bisa dikendalikan pergerakannya pake remote control mulai dari ekspresi wajah sampai pergerakan tangan dan kaki. Animatronik ini banyak digunakan pada wahana permainan seperti rumah hantu (yang jadi hantu hantunya), industri perfilman. Kebanyakan yang menggunakan teknik animatronik pada perfilman adalah hewan atau makhluk yang langka. perangkat kendali yang cukup praktis dan banyak digunakan adalah mikrokontroler yaitu sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan dapat menyimpan program didalamnya. Kelebihan utama mikrokontroler ialah tersedianya RAM dan peralatan I/O pendukung sehingga memiliki ukuran yang sangat ringkas dan lebih leluasa untuk dihubungkan dan melakukan pengendalian terhadap perangkat lain.*

**Kata Kunci:** *Arduino Uno, Mikrokontroller, Animatronic, robot, Balance.*

## **ABSTRACT**

*Animatronic derived from the animation and electronic. Its main characteristic is the use of robots, electronic smelling and using some sort of a mechanical doll. Can be controlled like a puppet movements use the remote control from the facial expression to the movement of the hands and feet. This animatronic widely used on rides like a ghost house (whose ghost haunted), the film industry. Most who use the technique of animatronics in film is a rare animal or creature. a control device that is practical and widely used is a microcontroller that is a chip that serves as an electronic control circuit and can store programs therein. The major advantage is the availability of microcontroller RAM and I / O support that has a very compact size and more flexibility to connect and exercise control over other devices.*

**Keywords:** *Arduino Uno microcontroller, Animatronic, robots, Balance.*

## KATA PENGANTAR



Alhamdulillahirobbil'alamin, puji dan syukur Penulis panjatkan atas kehadiran Allah SWT, karena berkat rahmat dan hidayah Nya penulis dapat menyelesaikan Skripsi ini dengan baik dan tepat pada waktunya.

Skripsi ini merupakan salah satu syarat untuk menyelesaikan Program Pendidikan Strata Satu ( S1 ) Teknik Informatika di Universitas Potensi Utama Medan. Skripsi ini disusun dengan judul “ **Perancangan Robot Penyeimbang Menggunakan Sensor Jarak Berbasis Android** “

Dalam penyusunan Skripsi dari awal hingga akhir penulis telah banyak mendapat bantuan, dukungan serta bimbingan dari berbagai pihak, oleh karena itu penulis mengucapkan terima kasih yang sebesar - besarnya kepada :

1. Bapak Edy Victor Haryanto, M.Kom, selaku Pembimbing I yang telah begitu banyak memberikan bimbingan dan petunjuk serta meluangkan waktunya selama menyelesaikan Skripsi ini.
2. Bapak Iwan Fitrianto Rahmad, M.Kom, selaku Pembimbing II yang telah begitu banyak memberikan masukan dalam menyelesaikan Skripsi ini.
3. Ibu Hj. Nuriandy, B.A, Selaku Pembina Yayasan Potensi Utama Medan.
4. Bapak Bob Subhan Riza, ST, M.Kom, selaku Ketua Yayasan Potensi Utama Medan
5. Ibu Rika Rosnelly, SH, M.Kom, selaku Rektor Universitas Potensi Utama.

6. Ibu Lili Tanti, M.kom, selaku Wakil Rektor I Universitas Potensi Utama.
7. Ibu Ratih Puspasari, M.Kom, selaku Dekan Fakultas Teknik dan Ilmu Komputer Universitas Potensi Utama.
8. Bapak Budi Triandi, M.Kom, selaku Ketua Program Studi Teknik Informatika Universitas Potensi Utama.
9. Seluruh Dosen, Staff, Karyawan dan Karyawati Universitas Potensi Utama.
10. Kepada Ayah dan Ibu tercinta yang selalu memberikan kasih sayang, semangat, Do'a, nasehat, serta dukungan moril untuk penulis sampai saat ini.
11. Teman–teman Perkuliahan di Universitas Potensi Utama, khususnya TI A, TI B, dan TI C Malam Angkatan 2012

Dalam penulisan Skripsi ini penulis menyadari masih terdapat kekurangan serta kelemahan dan masih jauh dari sempurna, baik dalam isi maupun penyajian materi. Oleh karena itu, penulis mengharapkan kritik dan saran yang bersifat membangun demi penyempurnaan serta kemajuan yang akan datang sangatlah penulis harapkan.

Akhirnya penulis berharap semoga Skripsi ini dapat diambil manfaatnya dan dapat digunakan sebagai mana mestinya.

Medan, 17 September 2016  
Penulis,

**(Muhammad Nazri)**  
**1110000271**

## DAFTAR ISI

<b>KATA PENGANTAR</b> .....	<b>i</b>
<b>DAFTAR ISI</b> .....	<b>iv</b>
<b>DAFTAR GAMBAR</b> .....	<b>vii</b>
<b>DAFTAR TABEL</b> .....	<b>ix</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>x</b>
<b>BAB IPENDAHULUAN</b> .....	<b>1</b>
I.1. LatarBelakang .....	1
I.2. RuangLingkupPermasalahan .....	3
I.2.1. IdentifikasiMasalah.....	3
I.2.2. RumusanMasalah.....	3
I.2.3. Batasanmasalah.....	4
I.3. Tujuan&Manfaat.....	4
I.3.1. Tujuan .....	4
I.3.2. Manfaat .....	5
I.4. MetodologiPenelitian.....	5
I.6. SistematikaPenulisan .....	6
<b>BAB IILANDASAN TEORI</b> .....	<b>8</b>
II.1. Pengertian Perancangan.....	8
II.2. Pengertian Dasar Sistem Kendali .....	9
II.3. Pengertian Sensor .....	11
II.4. Balancing Robot .....	12
II.5. Mikrokontroler ATMega 328 .....	13
II.5.1. Fitur AVR ATMega 328 .....	14
II.5.2. <i>Arsitektur Mikrokontroler ATMega 328</i> .....	15
II.5.3. <i>Konfigurasi Pin ATMega 328</i> .....	17
II.6. <i>Arduino Uno</i> .....	19

II.7. Bluetooth .....	20
II.8. LCD 16x2 .....	21
II.9. MPU6050 .....	23
II.10. Sensor Sharp GP2D12 .....	24
II.11. Regulator Tegangan .....	25
II.12. Bahasa Pemrograman .....	26
II.12.1. Bahasa Pemrograman C .....	26
II.12.2. Pemrograman IDE Arduino Uno .....	26
II.12.3. Android .....	28
II.12.3.1. Fitur-Fitur Android .....	29
II.12.3.2. Arsitektur Android .....	29
II.12.4. App Inventor .....	31
II.13. Flow Chart .....	32
<b>BAB III ANALISIS DAN PERANCANGAN .....</b>	<b>34</b>
III.1. Analisis Masalah .....	34
III.2. Strategi Pemecahan Masalah .....	35
III.3. Identifikasi Kebutuhan .....	36
III.3.1. Kebutuhan Perangkat Keras (Hardware) Perancangan Interface .....	36
III.3.2. Kebutuhan Perangkat Keras (Hardware) Perancangan Alat .....	36
III.3.3. Kebutuhan Perangkat Lunak (Software) .....	37
III.4. Diagram Blok Rangkaian .....	37
III.5. Perancangan .....	39
III.5.1. Perancangan Elektronik .....	39
III.5.1.1. Sistem Minimum Rangkaian Mikrokontroler ATmega 328 .....	39
III.5.1.2. Rangkaian Driver Motor L298 .....	41
III.5.1.3. Rangkaian Bluetooth HC-05 .....	41
III.5.1.4. Rangkaian Sensor MPU6050 .....	42
III.5.2. Perancangan Mekanik (Hardware) .....	43
III.5.3. Perancangan Perangkat Lunak (Software) .....	44
III.6. Flow Chart .....	45

III.6.1.Flow Chart Perangkat Robot Keseimbangan.....	45
III.6.2.Flow Chart Interface .....	48
<b>BAB IV HASIL DAN UJI COBA .....</b>	<b>50</b>
IV.1. Tampilan Hasil.....	50
IV.2. Pelaksanaan Pengujian Rangkaian Alat.....	50
IV.3.Tampilan Hasil Perangkat.....	51
IV.4. Pengujian Software .....	52
IV.5. Pengujian Hardware.....	64
IV.6.Uji Coba Perangkat .....	65
IV.6.1.Pengujian Rangkaian Mikrokontroler Arduino Uno .....	65
IV.6.2.Pengujian Rangkaian Regulator Tegangan .....	66
IV.6.3.Pengujian Rangkaian Sensor Jarak Sharpgp GP11 .....	67
IV.6.4.Pengujian Transmisi Data Bluetooth HC-05.....	70
IV.6.5.Penguji Analisa Perangkat Keseluruhan .....	72
IV.7.Kelebihan dan Kekurangan.....	75
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>77</b>
V.1. Kesimpulan .....	77
V.2. Saran.....	78

## **DAFTAR PUSTAKA**

## **LAMPIRAN**

## DAFTAR GAMBAR

Gambar II.1. Mikrokontroler ATmega328 .....	15
Gambar II.2. Tampilan Arsitektur ATmega328 .....	17
Gambar II.3. Minimum Sistem Arduino R3 .....	19
Gambar II.4. Bentuk Fisik Modul Bluetooth HC-06 .....	21
Gambar II.5. Bentuk Fisik LCD 16x2.....	22
Gambar II.6. Modul GY521 MPU-6050.....	23
Gambar II.7. Sensor Sharp GP2D12 .....	24
Gambar II.8. Bentuk Fisik dari Regulator 78xx.....	25
Gambar II.9. Tampilan Arduino IDE.....	27
Gambar II.10. Arsitektur Sistem Operasi Android .....	30
Gambar II.11. Tampilan Awal App Inventor .....	32
Gambar III. 1. Diagram Blok Rangkaian .....	38
Gambar III. 2. Skematik Rangkaian Mikrokontroler ATmega328 Pada Arduino Uno .....	40
Gambar III. 3. Skematik rangkaian Driver Motor L298 .....	41
Gambar III. 4. Skematik Pemasangan Bluetooth HC-05 Pada Arduino Uno .....	42
Gambar III. 5. Skematik Pemasangan MPU6050 Pada Arduino Uno .....	43
Gambar III. 6. Perancangan Mekanik .....	44
Gambar III. 7. Desain Tampilan Pada Smartphone Android .....	45
Gambar III. 8. Flowchart Rangkaian Robot.....	46
Gambar III. 9. Flowchart Interface .....	48
Gambar IV. 1. Perangkat Keseluruhan .....	52
Gambar IV. 2. Tampilan Arduino 1.6.9 .....	53
Gambar IV. 3 Pengaturan Dan Pemilihan Board Arduino.....	54
Gambar IV. 4. Pengaturan Port USB Pada Software Arduino 1.6.9.....	54
Gambar IV. 5. Proses Upload Program Software Arduino 1.6.9.....	62
Gambar IV. 6. Perancangan Software Aplikasi Pengendali .....	63
Gambar IV. 7. Blok Program Software Aplikasi Pengendali .....	64
Gambar IV. 8. Hardware Mekanik Dan Elektronik .....	65

Gambar IV. 9. Grafik Hasil Perbandingan Tegangan Normal Dengan Regulator 5VDC.....	67
Gambar IV. 10. Listing Program Pengujian Sharpgp GP11 .....	68
Gambar IV. 11. Grafik Hasil Perbandingan Sensor Jarak Sharpgp GP11 Terhadap Penggaris .....	70
Gambar IV. 12. Kondisi Normal Robot Penyeimbang .....	72
Gambar IV. 13. Kondisi Smartphone Menjalankan Aplikasi .....	73
Gambar IV. 14. Kondisi Smartphone Melakukan Koneksi .....	73
Gambar IV. 15. Kondisi Smartphone Berhasil Melakukan Koneksi .....	74
Gambar IV. 16. Robot Bergerak Menyesuaikan Perintah (1).....	74
Gambar IV. 17. Robot Bergerak Menyesuaikan Perintah (2).....	75

## DAFTAR TABEL

Tabel II.1. Tabel Simbol Flowchart .....	33
Tabel III.1. Konfigurasi Pin Pada Arduino .....	40
Tabel IV.1. Hasil Pengujian Regulator Tegangan 5VDC .....	66
Tabel IV.2. Hasil Pengujian Sensor Sharpgp GP11 Terhadap Penggaris.....	69
Tabel IV.3. Hasil Pengujian Jarak Transmisi Pada Halangan .....	70
Tabel IV.4.s Hasil Pengujian Jarak Transmisi Dengan Halangan .....	71

## **DAFTAR LAMPIRAN**

Lampiran-1	Listing Program
Lampiran-2	Surat Pengajuan Judul Skripsi
Lampiran-3	Formulir Pendaftaran Judul Skripsi
Lampiran-4	Surat Pernyataan Kesiadaan Pembimbing I
Lampiran-5	Surat Pernyataan Kesiadaan Pembimbing II
Lampiran-6	Jadwal Bimbingan Skripsi
Lampiran-7	Formulir Pendaftaran Seminar Hasil Skripsi
Lampiran-8	Formulir Pendaftaran Sidang Skripsi



## **BAB I**

### **PENDAHULUAN**

# **BAB I**

## **PENDAHULUAN**

### **I.1. Latar Belakang**

Teknologi adalah cara untuk mendapatkan suatu kualitas yang lebih baik, lebih mudah, lebih murah, lebih cepat dan lebih menyenangkan. Salah satu teknologi yang berkembang pesat pada saat ini adalah teknologi di bidang Robot. Robot berguna untuk membantu manusia dalam melakukan pekerjaan tertentu, misalnya melakukan pekerjaan yang memerlukan ketelitian tingkat tinggi, beresiko tinggi dan pekerjaan yang memerlukan tenaga besar.

Perkembangan robotika di tanah air selama beberapa tahun terakhir ini terus mengalami peningkatan. Peserta lomba robot yang secara tetap diadakan tiap tahun mulai bertambah khususnya dikalangan mahasiswa. Di tingkat nasional ada Kontes Robot Cerdas Indonesia yang diselenggarakan oleh Direktorat Perguruan Tinggi Indonesia dan pada tingkat internasional ada ROBOCON. Ternyata perkembangan robotika tidak hanya di kalangan mahasiswa, tetapi juga pada kalangan SD, SMP maupun SMA. Robot yang digunakan adalah LEGO ROBOTIC yaitu sebuah produk lego yang bisa dirangkai menjadi sebuah robot dan dapat diprogram dari komputer. LEGO ROBOTIC ini memadukan kemampuan mekanika dan kemampuan *programming* untuk memecahkan berbagai masalah.

Melihat keadaan ini, memperkenalkan pada khalayak bahwa robot juga bisa digunakan seperti simulasi atau *prototype* untuk dilakukan perancangan dan penganalisaan. Contoh sederhana yang dapat disimulasikan dengan robot adalah

*Robot balancing* atau robot penyeimbang merupakan robot beroda dua yang dapat menyeimbangkan diri. Saat ini kebanyakan sistem keseimbangan telah diterapkan pada robot yang dapat mengudara, tetapi sedikit yang diterapkan pada robot di daratan. Untuk mengatasi serta memperluas perkembangan teknologi robotik di daratan tersebut, maka penulis berinisiatif untuk merancang sebuah robot yang dapat menyeimbangkan diri dan dapat dikendalikan dengan *smartphone* android melalui *bluetooth* sebagai penghubung. Serta akan muncul cara baru dari kemampuan bermanuver dan mobilitas dalam aplikasi robotik di daratan, sehingga dapat menjadi acuan dalam perkembangan sistem transportasi yang ramah lingkungan dan modern.

Pada robot ini menggunakan mikrokontroler arduino, sensor *gyroscope* dan dua buah roda, dimana roda tersebut berada pada sebelah kanan dan kiri robot yang dapat menyeimbangkan diri sendiri.

Konsep robot penyeimbang didasarkan pada teori pendulum terbalik. Sebuah sistem kontrol yang sesuai dibutuhkan untuk mengontrol sistem sehingga seimbang dan stabil. Tujuan utama dari perancangan robot ini adalah untuk menjaga robot dalam keadaan tegak lurus. Meskipun di dunia robotika *Robot Balancing* sudah banyak dikenal, namun masih banyak yang dapat ditambahkan pada robot penyeimbang ini, dikarenakan saat ini robot penyeimbang hanya dibuat ada yang tanpa sensor jarak akan tetapi sudah menggunakan sistem android sebagai pengontrolnya dan ada juga robot penyeimbang yang dibuat dengan menggunakan sensor jarak akan tetapi tidak berbasis android. Berdasarkan latar

belakang di atas maka penulis akan mengangkat sebuah judul “**Perancangan Robot Penyeimbang Menggunakan Sensor Jarak Berbasis Android**”.

## **I.2. Ruang Lingkup Permasalahan**

### **I.2.1. Identifikasi Masalah**

Adapun identifikasi masalah dalam penulisan skripsi ini adalah:

1. Perancangan robot *balancing* sudah banyak yang membuat dan sudah populer dikalangan umum.
2. Dibutuhkan cara agar dapat mempertahankan kondisi tegak 90 derajat.
3. Dibutuhkan kalibrasi pada sensor jarak agar robot tidak menabrak dinding/halangan.
4. Dibutuhkan kalibrasi robot dengan kontroler agar dapat bergerak pada arah yang diharapkan.

### **I.2.2. Perumusan Masalah**

Permasalahan yang dapat dirumuskan dalam pengerjaan skripsi ini adalah sebagai berikut :

1. Bagaimana merancang konstruksi robot yang sanggup menyeimbangkan dirinya sendiri dengan benar?
2. Bagaimana sensor jarak membaca dinding/halangan dengan benar?
3. Bagaimana cara mensinkronkan robot, sensor, dan kontroler?
4. Bagaimana merancang aplikasi yang terhubung dengan robot penyeimbang yang berjalan pada *smartphone* android menggunakan komunikasi *bluetooth*?

### **I.2.3. Batasan Masalah**

Dalam penulisan skripsi ini dibatasi permasalahannya sebagai berikut :

1. Komunikasi *bluetooth* menggunakan *module bluetooth* HCL-05 atau sejenisnya dengan jarak robot dan kontroler maksimum 10 meter.
2. Perancangan ini menggunakan mikrokontroler Arduino.
3. Arduino yang digunakan adalah Arduino UNO R3 atau Arduino yang dirancang sendiri (*handmade*) atau sejenisnya.
4. *Software* yang digunakan adalah *Arduino IDE* sebagai pemrograman bahasa C untuk mikrokontroler.
5. Menggunakan LCD 16x2 untuk menampilkan informasi pada perangkat dan Penanda halangan hanya berupa *buzzer*
6. Sensor jarak yang digunakan merupakan Ultrasonik (PING))) atau *sharpgp* GP11 dan robot hanya membaca halangan berupa dinding ataupun objek lainnya.
7. Target OS android minimum yang dapat menjalankan aplikasi ini adalah OS Android 4.0 (*Ice Cream Sandwith*).
8. Perancangan perangkat ini dapat menggunakan beberapa sensor tambahan, seperti sensor jarak yang tidak dijelaskan secara terperinci.

## **I.3. Tujuan Dan Manfaat**

### **I.3.1. Tujuan**

Tujuan yang ingin dicapai melalui penulisan skripsi ini adalah sebagai berikut :

1. Untuk merancang robot berbasis mikrokontroler yang dapat menyeimbangkan diri dan menghindari objek.
2. Melakukan analisa kendali motor berdasarkan data sensor.
3. Robot yang dapat membaca data dari sensor untuk menyeimbangkan diri.
4. Untuk mengetahui prinsip kerja secara umum dari sistem robot penyeimbangan yang dirancang.
5. Melakukan koneksi antara perangkat mikrokontroler (Arduino) dengan *smartphone* android melalui *bluetooth* dan menganalisanya.
6. Melakukan analisa dari hasil pengujian.

### **I.3.2. Manfaat Penelitian**

Manfaat yang diharapkan dari penulisan skripsi ini adalah :

1. Alat serta sistem yang telah dibuat dapat diaplikasikan dalam kehidupan sehari-hari, terutama di bidang hiburan dan edukasi pengenalan mikrokontroler dan penggunaan aplikasi android.
2. Memahami cara kerja sensor *gyroscope* dan sensor jarak yang digunakan pada *balancing* robot dan mengetahui sistem pengendalian robot menggunakan *smartphone* berbasis android.
3. Perancangan ini dapat dikembangkan tidak hanya pada mengendalikan robot kecil, tetapi dapat dikembangkan sebagai sistem yang lebih besar seperti *segway* dan sebagainya.
4. Dengan adanya sistem ini dapat membantu dalam ilmu pengetahuan, khususnya dalam bidang Mekatronika, Teknologi Informasi dan Ilmu Komputer.

#### **I.4. Metodologi Penelitian**

Untuk dapat mengimplementasikan sistem di atas, maka secara garis besar digunakan beberapa metode sebagai berikut:

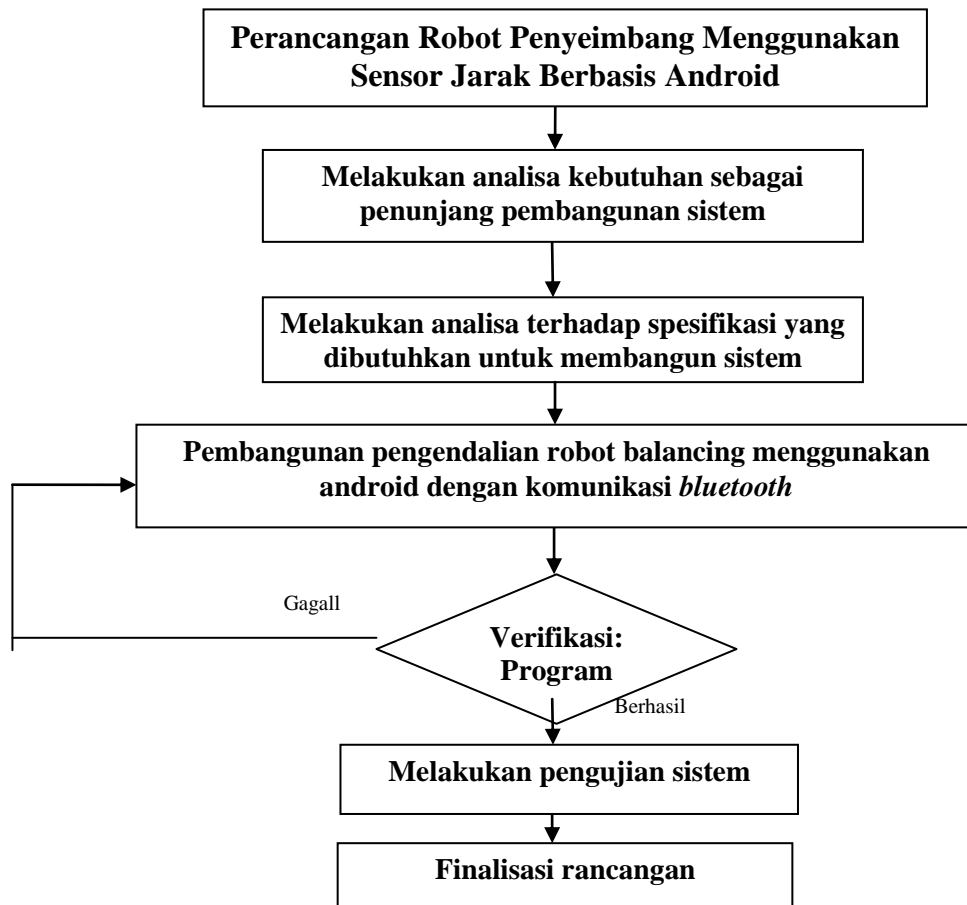
1. Studi Literatur, dengan cara mempelajari buku-buku acuan dan literatur yang berhubungan dengan materi dalam penulisan skripsi.
2. Pengumpulan Data, yaitu mengumpulkan informasi dan mempelajari komunikasi antara android dengan alat melalui *bluetooth*.
3. Analisa Permasalahan, untuk mengetahui dan menentukan batasan-batasan sistem sehingga dapat menentukan cara yang paling efektif dalam penyelesaian permasalahan.
4. Perancangan Alat, setelah menganalisa permasalahan, selanjutnya dilakukan pengumpulan data dan perancangan alat yang telah ditetapkan.
5. Implementasi alat, membuat alat berdasarkan rancangan alat yang telah dibuat sesuai dengan data yang ada.
6. Uji coba alat, menguji alat yang telah dibuat, untuk mengetahui letak kesalahan dan memperbaikinya.
7. Dokumentasi, membuat laporan dari semua pengerjaan yang telah dilakukan.

##### **I.4.1. Analisis Tentang Sistem Yang Ada**

Penulis mempelajari dasar teori dari berbagai literatur sebagai dasar untuk melakukan selanjutnya.

## 1. Prosedur Perancangan

Langkah – langkah yang diperlukan untuk mencapai tujuan perancangan dapat dilihat pada gambar di bawah ini:



Gambar I.1. Prosedur Perancangan

## 2. Analisis Kebutuhan

Setelah melalui tahap prosedur rancangan, maka tahap selanjutnya dibutuhkan data - data analisa Dalam membuat perancangan *balancing* robot menggunakan android dengan komunikasi *bluetooth* ini, diuraikan spesifikasi dan desain yang digunakan berupa perangkat keras (*hardware*) dan perangkat lunak (*software*).

### 3. Spesifikasi dan Desain

Dalam membuat skripsi ini, diuraikan spesifikasi dan desain yang digunakan berupa perangkat keras (*hardware*) dan perangkat lunak (*software*).

#### a. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan antara lain ;

1. *Core i3 ; Processor 2.53 GHz*
2. *Hard disk : 320 GB*
3. *RAM 2 GB*
4. *Monitor LCD 14"*
5. *Keyboard dan Mouse.*
6. *Smartphone Android*

#### b. Perangkat lunak (*Software*)

*Software* yang digunakan untuk membuat perancangan ini antara lain :

1. Sistem operasi *Windows 7 32bit.*
2. *Arduino IDE 1.6.5*
3. *App Inventor*
4. Sistem Operasi Android min. *Ice Cream Sandwich*

### 4. Implementasi dan Verifikasi

Setelah analisis dan perancangan, maka perlu dilakukan implementasi atau uji coba terhadap robot yang telah selesai dibuat. Hal ini dilakukan untuk pengembangan atau perbaikan pada robot tersebut apakah sudah bekerja sesuai dengan rancangan. Langkah-langkah dalam perancangan robot penyeimbang menggunakan sensor jarak berbasis android yaitu:

1. Pengumpulan data didapatkan dari berbagai sumber diantaranya: buku-buku mikrokontroler, internet dan sumber lainnya yang sesuai.
2. Perancangan elektornik dan mekanik robot.
3. Perancangan komunikasi robot *balancing* dengan *smartphone android* menggunakan *bluetooth*.
4. Perancangan *software interface* pada android.

## **5. Validasi**

Pada tahap ini dilakukan pengujian *balancing* robot menggunakan android dengan komunikasi *bluetooth* secara menyeluruh, meliputi pengujian fungsional dan pengujian ketahanan sistem. Pengujian fungsional dilakukan untuk mengetahui bahwa sistem *balancing* robot dapat bekerja dengan baik sesuai dengan prinsip kerjanya. Validasi dilakukan dengan pentransferan atau *download script / coding* ke Arduino serta melakukan pengujian.

## **6. Pengujian / Uji Coba**

Pada tahap ini dilakukan uji coba sistem pengendalian *balancing* robot menggunakan android dengan komunikasi *bluetooth*.. Apakah komunikasi *bluetooth* antara aplikasi android dengan *balancing* robot tersebut bisa spesifik dalam mengirim data, menguji ketahanan alat apakah dapat dipergunakan dengan sebaik mungkin.

## **7. Finalisasi**

Pada tahapan ini adalah tahapan hasil dari robot yang sudah dirancang dan berjalan sesuai rencana.

## **I.5. Sistematika Penulisan**

Adapun sistematika penulisan skripsi ini, adalah sebagai berikut :

### **BAB I : PENDAHULUAN**

Berisikan tentang latar belakang, ruang lingkup permasalahan, tujuan dan manfaat, metodologi penelitian dan sistematika penulisan.

### **BAB II : TINJAUAN PUSTAKA**

Berisikan tentang Teori dan metode yang berhubungan dengan topik pokok yang dibahas.

### **BAB III : ANALISIS DAN PERANCANGAN**

Berisikan tentang analisis sistem yang berjalan, evaluasi sistem yang berjalan dan desain *balancing* robot menggunakan android dengan komunikasi *bluetooth*.

### **BAB IV : HASIL DAN UJI COBA**

Berisikan tentang tampilan hasil pengendali *balancing* robot menggunakan android dengan komunikasi *bluetooth*.

### **BAB V : KESIMPULAN DAN SARAN**

Berisikan tentang kesimpulan dan saran dari hasil penelitian skripsi *balancing* robot menggunakan android dengan komunikasi *bluetooth*.



## **BAB II**

### **LANDASAN TEORI**

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Pengertian Perancangan**

Desain atau perancangan dalam pengembangan perangkat lunak merupakan upaya untuk mengkonstruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional memenuhi target, memenuhi kebutuhan secara implisit dan eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat. (Rosa A.S, M. Shalahuddin ; 2011: 21).

Desain atau perancangan adalah mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem *fokus* pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan. (Rosa A.S, M. Shalahuddin ; 2011: 25).

Desain atau perancangan perangkat lunak adalah proses langkah yang *fokus* pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan perangkat lunak ke representasi desain agar dapat di implementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu di dokumentasikan. (Rosa A.S, M. Shalahuddin ; 2011: 27).

Perancangan adalah proses menuangkan ide dan gagasan berdasarkan teori-teori dasar yang mendukung. Proses perancangan dapat dilakukan dengan

cara pemilihan komponen yang akan digunakan, mempelajari karakteristik dan data fisiknya, membuat rangkaian skematik dengan melihat fungsi-fungsi komponen yang dipelajari, sehingga dapat dibuat alat yang sesuai dengan spesifikasi yang diharapkan. (<http://elib.unikom.ac.id/download.php?id=143047>: dikunjungi pada tanggal 28 Juli 2016).

## **II.2. Pengertian Dasar Sistem Kendali**

Sistem kendali merupakan suatu sistem yang keluarannya dikendalikan pada suatu nilai tertentu atau untuk mengubah beberapa ketentuan yang telah ditetapkan dari masukan ke sistem. Untuk merancang suatu sistem yang dapat merespon suatu perubahan tegangan dan mengeksekusi perintah berdasarkan situasi yang terjadi, maka diperlukan pemahaman tentang sistem kendali (*control system*). Sistem kendali merupakan suatu kondisi dimana sebuah perangkat (*device*) dapat dikendalikan sesuai dengan perubahan situasi.

Kendali (pengendalian) adalah sebuah proses atau upaya untuk mencapai tujuan. Dengan demikian, sistem kendali adalah kombinasi dari beberapa komponen (*subsistem*) yang bekerja secara sinergi dan terpadu untuk memperoleh hasil yang diinginkan (tujuan). Sebagai contoh sederhana dan akrab dengan aktivitas sehari-hari dari konsep kendali adalah saat mengendarai kendaraan. Tujuan yang diinginkan dari proses tersebut adalah berjalannya kendaraan pada lintasan (*track*) yang diinginkan. Ada beberapa komponen yang terlibat di dalamnya, misalnya pedal gas, *speedometer*, mesin (penggerak), rem, dan pengendara. Sistem kendali berkendara berarti kombinasi dari komponen-komponen tersebut yang menghasilkan berjalannya kendaraan pada lintasan yang

diinginkan. Ketika jalan lengang dan aturan memperbolehkan, pengendara mempercepat laju kendaraan dengan membuka pedal gas. Demikian pula, jika ada kendaraan lain di depan atau lampu penyeberangan berwarna merah maka pengendara menginjak rem dan menurunkan kecepatannya. Semua upaya itu dilakukan untuk mempertahankan kendaraan pada lintasan yang diinginkan.

Contoh lain dapat disebutkan berupa proses memindahkan barang oleh tangan kita. Pada proses tersebut, tujuannya adalah posisi atau letak barang yang diinginkan. Komponennya berupa tangan (dalam hal ini tentunya dengan otot tangan), mata, dan otak sebagai pengendali. Pada saat tangan bergerak untuk memindahkan barang, mata akan menangkap informasi tentang posisi pada saat itu. Informasi tersebut diproses oleh otak untuk disimpulkan apakah posisinya sudah benar atau tidak. Selanjutnya, apabila posisinya masih belum tercapai maka otak akan memerintahkan otot tangan untuk bergerak memindahkan barang ke posisi yang diinginkan. Secara umum dapat dikatakan semua proses yang terjadi di alam pada hakikatnya adalah sebuah sistem kendali.

Ada beberapa istilah yang biasanya dipakai dalam pembahasan sistem kendali, yaitu:

1. *Plant*, *Plant* berarti objek yang dikendalikan, misalnya pesawat terbang, reaktor kimia, tungku pembakaran, dan lain-lain.
2. *Input* (masukan), *Input* adalah sinyal yang masuk ke dalam sistem atau subsistem untuk selanjutnya diproses agar menghasilkan *output*.
3. *Output* (keluaran), *Output* berarti sinyal yang dihasilkan oleh proses yang terjadi dalam sistem atau subsistem.

4. *Command input* atau *reference input* (masukkan acuan) adalah tujuan dari proses yang terjadi dalam sistem kendali.
5. *Aktuator*, Subsistem ini berfungsi untuk menghasilkan *output* agar mampu menggerakkan *plant* ke arah yang diinginkan.
6. Pengendali (*controller*), Subsistem pengendali adalah “otak”nya sistem kendali yang berisi algoritma kendali sedemikian sehingga menghasilkan aturan-aturan pengendalian (*control law*).
7. *Sensor* (dan transduser), *Sensor* berfungsi mendeteksi keluaran yang terjadi untuk dibandingkan dengan masukan acuan. Nama *transduser* dipakai apabila dalam prakteknya melibatkan perubahan besaran.
8. *Komparator*, Subsistem ini berfungsi membandingkan dua sinyal yang masuk, yaitu menjumlahkan atau mengurangi.
9. *Disturbance* (gangguan) atau *Noise* (derau), Istilah ini dipakai untuk menamai sinyal yang masuk ke dalam sistem atau *plant*, tetapi bersifat “menggangu” pencapaian proses kendali ke arah tujuan yang diinginkan. (Asep Najmurokhman, 2010 : 1)

### **II.3. Pengertian Sensor**

Dalam rangkaian elektronika untuk keperluan pengukuran atau deteksi, diperlukan suatu bagian yang disebut sensor. Sensor berfungsi untuk menubah besaran yang bersifat fisis atau suhu, tekanan, berat, atau intensitas cahaya menjadi besaran listrik (tegangan atau arus listrik).

Sensor adalah alat untuk mendeteksi atau mengukur sesuatu yang digunakan untuk mengubah variasi mekanis, *magnetic*, panas, dan kimia menjadi

tegangan dan arus listrik. Dalam lingkungan sistem pengendali dan robotika, sensor memberikan kesamaan yang menyerupai mata, pendengaran, hidung, lidah yang kemudian akan diolah oleh kontroler sebagai otaknya. (Petruzella : 2010)

Sensor memiliki suatu ukuran yang disebut sensitivitas. Sensitivitas menunjukkan seberapa besar pengaruh perubahan nilai besaran fisis yang diukur oleh sensor terhadap keluaran dari sensor tersebut. Misalnya, sebuah sensor suhu yang tegangan keluarannya berubah 0,1 V jika terjadi perubahan suhu sebesar  $1^{\circ}\text{C}$ . Maka sensor suhu tersebut dapat dikatakan memiliki sensitivitas sebesar  $0,1\text{V}/1^{\circ}\text{C}$ . (Endra Pitowarno ; 2006 : 56)

Sensor yang baik memiliki ciri-ciri sebagai berikut:

1. Peka terhadap besaran yang akan diukur.
2. Tidak peka terhadap besaran lain yang tidak akan diukur.
3. Keberadaan sensor tidak mempengaruhi besaran yang akan diukur.

#### **II.4. Balancing Robot**

*Balancing* robot (robot penyeimbang) beroda dua merupakan suatu robot mobile yang memiliki dua buah roda disisi kanan dan kirinya yang tidak akan seimbang apabila tanpa adanya kontroler. Balancing robot ini merupakan pengembangan dari model pendulum terbalik (*inverted pendulum*) yang diletakkan di atas kereta beroda. Menyeimbangkan robot beroda dua memerlukan suatu metode kontrol yang baik dan handal untuk mempertahankan posisi robot dalam keadaan tegak lurus terhadap permukaan bumi tanpa memerlukan pengendali lain dari luar.

## II.5. Mikrokontroler ATmega328

Mikrokontroler adalah sebuah sistem mikroprosesor dimana didalamnya sudah terdapat *CPU*, *Read Only Memory (ROM)*, *Random Access Memory (RAM)*, *Input-Output*, *timer*, *interrupt*, *Clock* dan peralatan *internal* lainnya yang sudah saling terhubung dan terorganisasi dengan baik dalam satu *chip* yang siap dipakai (Heri Susanto, et al. 2013 : 3).

Dengan kata lain, 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. Mikrokontroler merupakan komputer didalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC TTL dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh mikrokontroler.

Mikrokontroler digunakan dalam produk dan alat yang dikendalikan secara otomatis, seperti sistem kontrol mesin, *remote controls*, mesin kantor, peralatan rumah tangga, alat berat, dan mainan. Dengan mengurangi ukuran, biaya, dan konsumsi tenaga dibandingkan dengan mendesain menggunakan mikroprosesor memori, dan alat *input output* yang terpisah, kehadiran mikrokontroler membuat kontrol elektrik untuk berbagai proses menjadi lebih ekonomis. Dengan penggunaan mikrokontroler ini maka :

1. Sistem elektronik akan menjadi lebih ringkas
2. Rancang bangun sistem elektronik akan lebih cepat karena sebagian besar dari sistem adalah perangkat lunak yang mudah dimodifikasi
3. Pencarian gangguan lebih mudah ditelusuri karena sistemnya yang kompak

Agar sebuah mikrokontroler dapat berfungsi, maka mikrokontroler tersebut memerlukan komponen eksternal yang kemudian disebut dengan sistem minimum. Untuk membuat sistem minimal paling tidak dibutuhkan sistem *clock* dan reset, walaupun pada beberapa mikrokontroler sudah menyediakan sistem *clock internal*, sehingga tanpa rangkaian eksternal pun mikrokontroler sudah beroperasi.

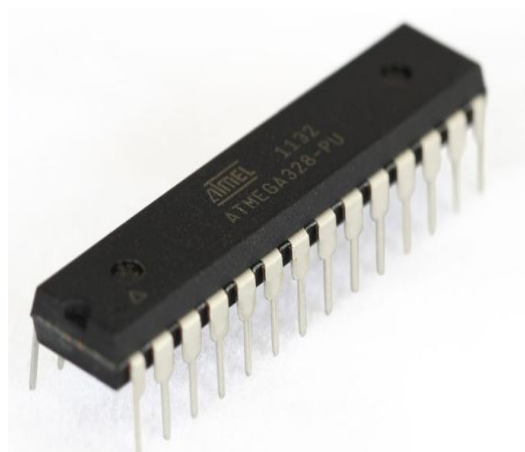
Yang dimaksud dengan sistem minimal adalah sebuah rangkaian mikrokontroler yang sudah dapat digunakan untuk menjalankan sebuah aplikasi. Sebuah IC mikrokontroler tidak akan berarti bila hanya berdiri sendiri. Pada dasarnya sebuah sistem minimal mikrokontroler AVR memiliki prinsip yang sama (Bernike Natalia Ginting, 2012 : 2).

### **II.5.1. Fitur AVR ATMega328**

ATMega328 adalah mikrokontroler keluaran dari atmel yang mempunyai arsitektur *Reduce Instruction Set Computer (RISC)* dimana setiap proses eksekusi data lebih cepat dari pada arsitektur *Completed Instruction Set Computer (CISC)*. Mikrokontroler ini memiliki beberapa fitur antara lain (Menurut Baaret, 2013 : 3)

1. 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus *clock*.
2. 32 x 8-bit *register* serba guna.

3. Kecepatan mencapai 16 *MIPS* dengan *clock* 16 MHz.
4. 32 KB *Flash memory* dan pada arduino memiliki *bootloader* yang menggunakan 2 KB dari *flash* memori sebagai *bootloader*.
5. Memiliki *EEPROM (Electrically Erasable Programmable Read Only Memory)* sebesar 1KB sebagai tempat penyimpanan data semi permanent.
6. Memiliki *SRAM (Static Random Access Memory)* sebesar 2KB.
7. Memiliki pin *I/O* digital sebanyak 14 pin 6 diantaranya *PWM (Pulse Width Modulation) output*.
8. *Master / Slave SPI Serial interface*.



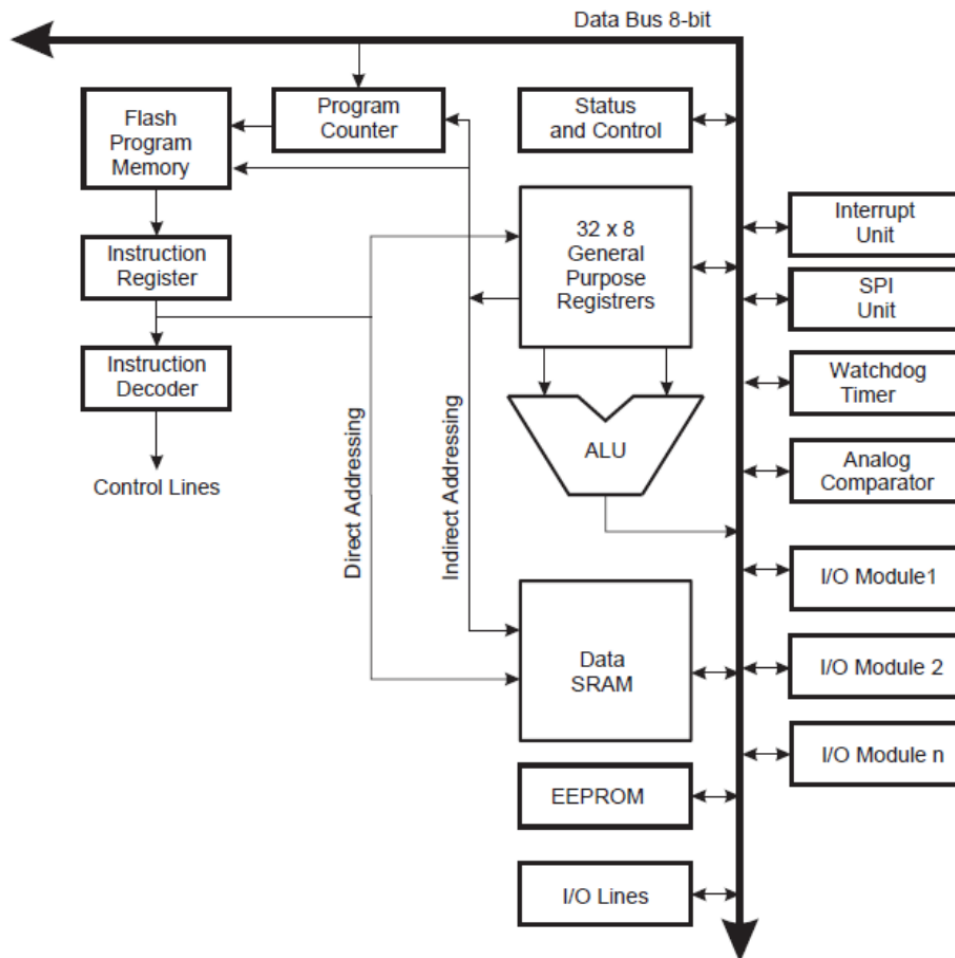
**Gambar II.1. Mikrokontroler ATmega328**

(Sumber: <http://www.protostack.com> dikunjungi pada tanggal 28 Juli 2016)

### II.5.2. Arsitektur Mikrokontroler ATmega328

Seluruh mikrokontroler yang diimplementasikan pada produk Arduino menggunakan ATmega Keluaran AVR. Salah satunya, seri ATmega328 dengan sejumlah fitur di antaranya *On-Chip System Debug*, 5 ragam tidur (*Mode Sleep*), 6 saluran *ADC* yang mendukung reduksi *derau*, ragam hemat daya (*Power-save Mode, Power-down*), dan *Standby Mode*. (Jazia Eko Istiyanto, 2014 : 6).

Mikrokontroler ATmega328 memiliki arsitektur *Harvard*, yaitu memisahkan memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan kerja dan *parallelism*. Instruksi-instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat satu instruksi dikerjakan instruksi berikutnya sudah diambil dari memori program. Konsep inilah yang memungkinkan instruksi – instruksi dapat dieksekusi dalam setiap satu siklus *clock*. 32 x 8-bit *register* serba guna digunakan untuk mendukung operasi pada *ALU* ( *Arithmetic Logic unit* ) yang dapat dilakukan dalam satu siklus. 6 dari *register* serbaguna ini dapat digunakan sebagai 3 buah *register pointer* 16-bit pada mode pengalamatan tidak langsung untuk mengambil data pada ruang memori data. Ketiga *register pointer* 16-bit ini disebut dengan *register X* ( gabungan R26 dan R27 ), *register Y* ( gabungan R28 dan R29 ), dan *register Z* ( gabungan R30 dan R31 ). Hampir semua instruksi AVR memiliki *format* 16-bit. Setiap alamat memori program terdiri dari instruksi 16-bit atau 32-bit. Selain *register* serba guna di atas, terdapat *register* lain yang terpetakan dengan teknik *memory mapped I/O* selebar 64 byte. Beberapa *register* ini digunakan untuk fungsi khusus antara lain sebagai *register control Timer/Counter*, Interupsi, *ADC*, *USART*, *SPI*, *EEPROM*, dan fungsi *I/O* lainnya. *Register – register* ini menempati memori pada alamat 0x20h – 0x5Fh. Berikut ini adalah tampilan *architecture* ATmega328 :



**Gambar II.2. Tampilan Arsitektur ATmega328**

(Sumber : Rizqi Ramadhan, 2012 : 5)

### II.5.3. Konfigurasi Pin ATmega328

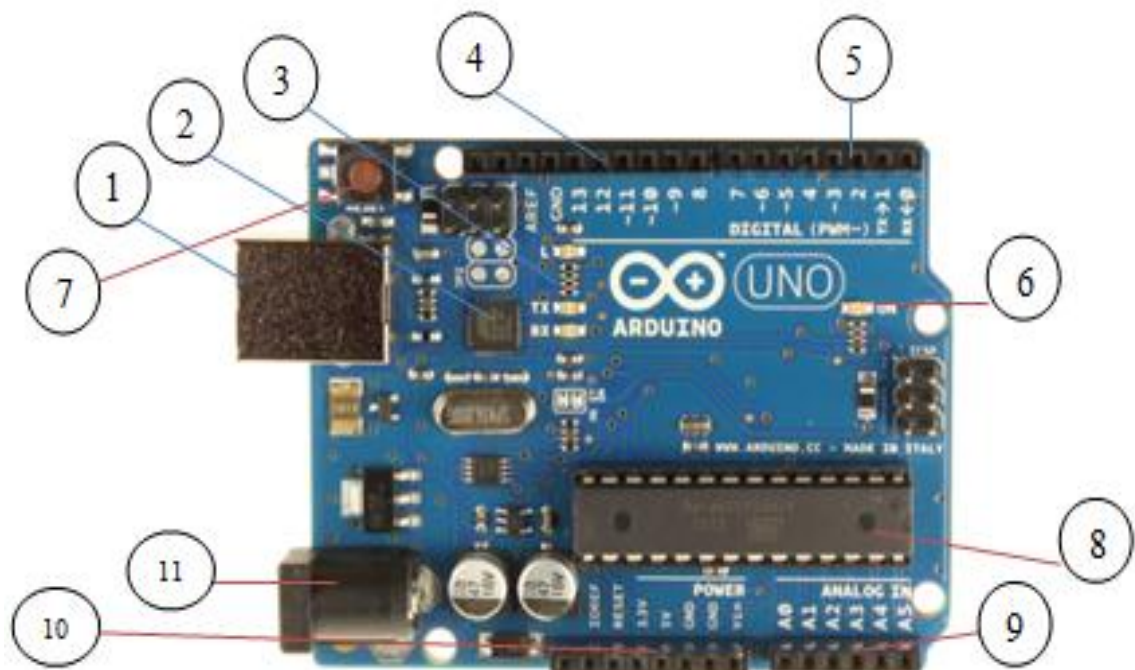
Masing-masing dari 14 pin digital arduino uno dapat digunakan sebagai masukan atau keluaran menggunakan fungsi `pinMode()`, `digitalWrite()` dan `digitalRead()`. Setiap pin beroperasi pada tegangan 5 volt. Setiap pin mampu menerima atau menghasilkan arus maksimum sebesar 40 mA dan memiliki resistor *pull-up internal* (diputus secara *default*) sebesar 20-30 KOhm. Sebagai tambahan, beberapa pin masukan digital memiliki kegunaan khusus yaitu (Jazi Eko Istiyanto, 2014 : 61):

1. Komunikasi serial: pin 0 (RX) dan pin 1 (TX), digunakan untuk menerima(RX) dan mengirim(TX) data secara serial.
2. External Interrupt: pin 2 dan pin 3, pin ini dapat dikonfigurasi untuk memicu sebuah interrupt pada nilai rendah, sisi naik atau turun, atau pada saat terjadi perubahan nilai.
3. *Pulse-width modulation (PWM)*: pin 3,5,6,9,10 dan 11, menyediakan keluaran *PWM* 8-bit dengan menggunakan fungsi *analogWrite()*.
4. *Serial Peripheral Interface (SPI)*: pin 10 (SS), 11 (MOSI), 12 (MISO) dan 13 (SCK), pin ini mendukung komunikasi SPI dengan menggunakan *SPI library*.
5. LED: pin 13, terdapat built-in LED yang terhubung ke pin digital 13. Ketika pin bernilai *HIGH* maka LED menyala, sebaliknya ketika pin bernilai *LOW* maka LED akan padam.

Arduino Uno memiliki 6 masukan analog yang diberi label A0 sampai A5, setiap pin menyediakan resolusi sebanyak 10 bit (1024 nilai yang berbeda). Secara *default* pin mengukur nilai tegangan dari *ground* (0V) hingga 5V, walaupun begitu dimungkinkan untuk mengganti nilai batas atas dengan menggunakan pin AREF dan fungsi *analogReference()*. Sebagai tambahan beberapa pin masukan analog memiliki fungsi khusus yaitu pin A4 (SDA) dan pin A5 (SCL) yang digunakan untuk komunikasi *Two Wire Interface (TWI)* atau *Inter Integrated Circuit (I2C)* dengan menggunakan *Wire library* yaitu (Jazi Eko Istiyanto, 2014 : 61).

## II.6. Arduino Uno

Arduino Uno R3 adalah papan mikrokontroler berdasarkan ATMEGA8. Arduino ini memiliki 14 digital *input / output* pin (dimana 6 dapat digunakan sebagai *output* PWM), 6 input analog, *osillator kristal* 16 MHz, koneksi USB, *jack* DC, header ICSP, dan tombol reset. *Board* ini berisi semua yang diperlukan untuk mendukung mikrokontroler, cukup menghubungkannya ke komputer dengan kabel USB atau kekuasaan itu dengan adaptor AC-DC atau baterai untuk memulai. Arduino Uno R3 berbeda dari semua papan sebelumnya yang tidak menggunakan *chip* FTDI *driver* USB-to-serial. Sebaliknya, Arduino ini memiliki fitur Atmega16U2 (Atmega8U2 hingga versi R2) diprogram sebagai konverter USB-to-serial.



**Gambar II.3. Minimum Sistem Arduino R3**

Sumber : [www.arduino.cc](http://www.arduino.cc)

Keterangan:

1. Port USB
2. IC Konverter Serial-USB (ATmega 8 U2)
3. Led untuk test output kaki D13
4. Kaki-kaki Input Output Digital (D8 – D13)
5. Kaki-kaki input Output Digital (D0 – D7)
6. LED Indikator catu daya
7. Tombol Reset
8. Mikrokontroler ATmega 8
9. Kaki-kaki input analog (A0 – A5)
10. Kaki-kaki catu daya (5V, GND)
11. Terminal Catudaya (6 – 9V)

## **II.7. Bluetooth**

*Bluetooth* sebuah teknologi komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz *unlicensed ISM (Industrial, Scientific and Medical)* dengan menggunakan sebuah *frequency hopping tranceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real-time* antara *host to host bluetooth* dengan jarak jangkauan layanan yang terbatas. *Bluetooth* sendiri dapat berupa *card* yang bentuk dan fungsinya hampir sama dengan card yang digunakan untuk *Wireless Local Area Network (WLAN)* dimana menggunakan frekuensi radio standar IEEE802.11, hanya saja pada *bluetooth* mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan *transfer* data yang lebih rendah.

Sistem *Bluetooth* terdiri dari sebuah *radio transceiver*, *baseband link Management dan Control*, *Baseband (processor core, SRAM, UART, PCM USB Interface)*, *flash* dan *voice code*. sebuah *link manager*. *Baseband link controller* menghubungkan perangkat keras radio ke *baseband processing* dan *layer protocol* fisik. *Link manager* melakukan aktivitas-aktivitas protokol tingkat tinggi seperti melakukan *link setup*, autentikasi dan konfigurasi. Bentuk fisik modul *Bluetooth HC-06* dapat dilihat pada gambar II.4 di bawah ini :



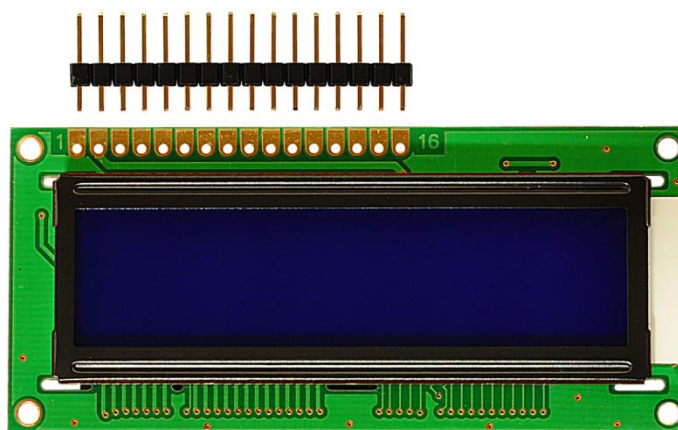
**Gambar II.4. Bentuk Fisik Modul Bluetooth HC-06**

Sumber : <http://www.botscience.net>

## II.8. LCD 16 x 2

LCD (*Liquid Crystal Display*) merupakan suatu jenis media tampilan yang menggunakan *crystal* cair sebagai penampil utama. LCD adalah salah satu perangkat penampil yang sekarang ini mulai banyak digunakan. LCD memanfaatkan silikon atau galium dalam bentuk kristal cair sebagai pemancar cahaya. Pada layar LCD, setiap matrik adalah susunan dua dimensi piksel yang dibagi dalam baris dan kolom. Dengan demikian setiap pertemuan baris dan kolom adalah sebuah LED terdapat sebuah bidang latar (*backplane*), yang

merupakan lempengan kaca bagian belakang dengan sisi dalam yang ditutupi oleh lapisan elektroda transparan. Dalam keadaan normal, cairan yang digunakan memiliki warna cerah. Daerah-daerah tertentu pada cairan akan berubah warnanya menjadi hitam ketika tegangan diterapkan antara bidang latar dan pola elektroda yang terdapat pada sisi dalam lempeng kaca bagian depan. Bentuk fisik LCD 16x2 dapat dilihat pada gambar II.5



**Gambar II.5. Bentuk Fisik LCD 16x2**

Sumber : 20 Aplikasi mikrokontroler ATmega8535 & ATmega8535 menggunakan Bascom-AVR, Afrie Setiawan

LCD yang digunakan adalah jenis LCD yang menampilkan data dengan 2 baris tampilan pada *display*. Keuntungan dari LCD ini adalah

1. Dapat menampilkan karakter ASCII, sehingga dapat memudahkan dalam pembuatan program tampilan.
2. Mudah dihubungkan dengan *port I/O* karena hanya menggunakan 8 bit data.
3. Ukuran modul yang proporsional.
4. Daya yang digunakan relatif sangat kecil.

## II.9. MPU6050

MPU6050 adalah *chip IC invense* yang di dalamnya terdapat sensor *accelerometer* dan *Gyroscope* yang sudah terintegrasi. Alasan menggunakan sensor ini adalah karena harganya relatif murah dimana sudah mendapatkan 2 sensor yang sudah terintegrasi.



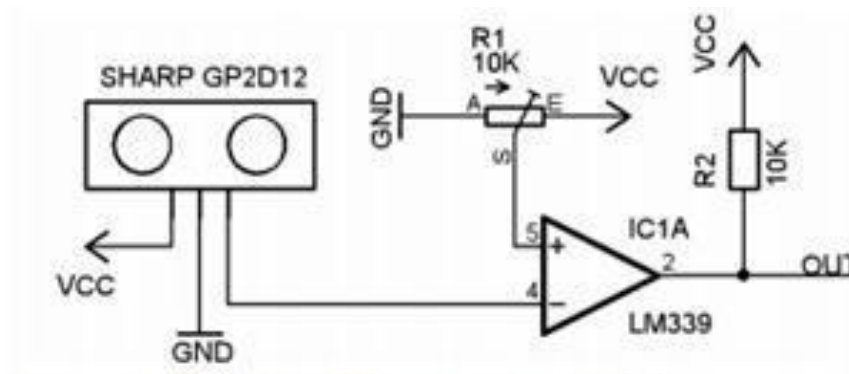
**Gambar II.6. Modul GY521 MPU-6050**  
Sumber : *Datasheet* GY521 MPU-6050

Berikut adalah spesifikasi dari Modul GY521 MPU-6050 ini :

1. Berbasis *Chip* MPU-6050
2. *Supply* tegangan berkisar 3-5V
3. *Gyroscope range* + 250 500 1000 2000 ° / s
4. *Acceleration range*:  $\pm 2 \pm 4 \pm 8 \pm 16$  g
5. *Communication standard* I2C
6. *Chip built-in* 16 bit AD *converter*, 16 bits data output
7. Jarak antar pin *header* 2.54 mm
8. Dimensi modul 20.3mm x 15.6mm

## II.10. Sensor Sharp GP2D12

Sensor sharp GP2D12 digunakan untuk membaca jarak. Sensor ini menggunakan prinsip pantulan sinar infra merah. Dalam aplikasi ini nilai tegangan keluran dari sensor yang berbanding terbalik dengan hasil pembacaan jarak dikomparasi dengan tegangan referensi komparator. Rangkaian sistem komparator pembacaan jarak dengan sensor sharp GP2D12 ini disajikan pada gambar berikut:



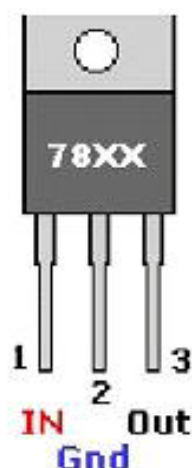
**Gambar II.7. Sensor Sharp GP2D12**  
 Sumber : *Datasheet* Sensor Sharp GP2D12

Prinsip kerja dari rangkaian komparator sensor sharp GP2D12 pada Gambar diatas adalah jika sensor mengeluarkan tegangan melebihi tegangan referensi, maka keluaran dari komparator akan berlogika rendah. Jika tegangan

referensi lebih besar dari tegangan sensor maka keluaran dari komparator akan berlogika tinggi. Selain menggunakan komparator, untuk mengakses sensor jarak sharp GP2D12 dapat dengan menggunakan prinsip ADC, atau dengan kata lain mengolah sinyal analog dari pembacaan sensor sharp GP2D12 ke bentuk digital dengan bantuan pemrograman. Untuk mengakses sensor ini dapat menggunakan fasilitas akses ADC.

### II.11. Regulator Tegangan

*Regulator* seri 7805 adalah *regulator* untuk mendapatkan tegangan keluaran sebesar +5 volt, sedangkan *regulator* seri 7812 adalah untuk mendapatkan tegangan keluaran sebesar +12 volt. Agar rangkaian *regulator* dengan *IC* tersebut dapat bekerja dengan baik, tegangan *input* harus lebih besar dari tegangan *output* *regulator*-nya. Bentuk Fisik dari *regulator* 78xx dapat dilihat pada gambar II.8. (Fredy Indra Oktaviansyah: 2011).



**Gambar II.8. Bentuk Fisik dari Regulator 78xx**

Sumber : Datasheet LM 78xx

## II.12. Bahasa Pemrograman

### II.12.1. Bahasa Pemrograman C

Struktur penulisan bahasa C secara umum terdiri atas empat blok, yaitu:

1. *Header*,
2. Deklarasi konstanta global dan atau variabel,
3. Fungsi dan atau prosedur (bisa di bawah program utama),
4. Program utama.

Secara umum, pemrograman C paling sederhana dilakukan dengan hanya menuliskan program utamanya saja, yaitu:

```

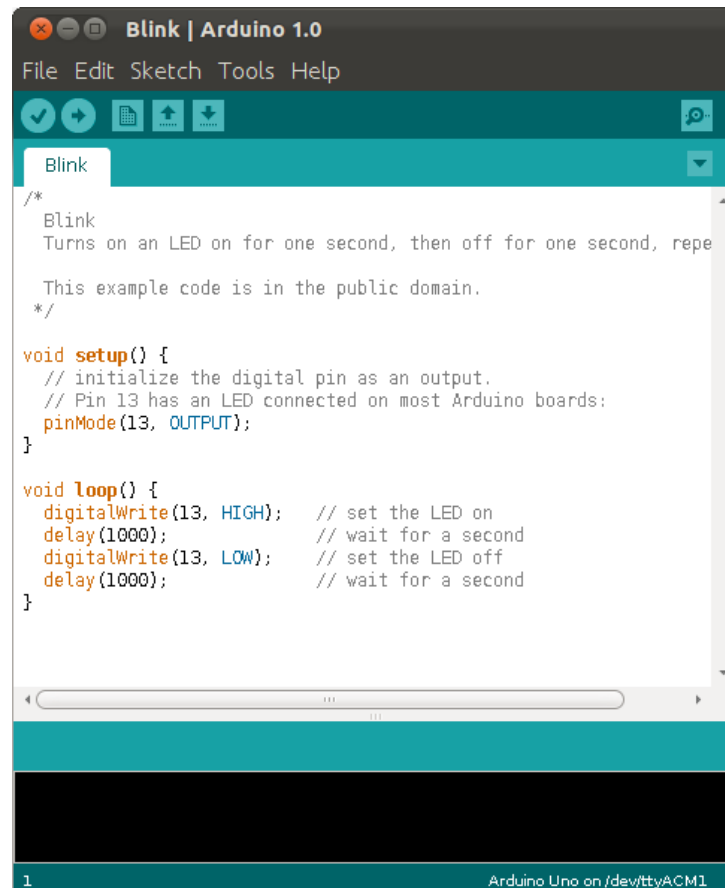
/* fungsi utama */
void main()
{
    Statemen-statemen;
}
/* fungsi-fungsi lain yang ditulis oleh pemrogram komputer */
Fungsi_fungsi_lain()
{
    Statemen-statemen;
}

```

(M. Ary Heryanto, ST dan Ir. Wisnu Adi P. (2008: 8))

### II.12.2. Pemrograman *IDE* Arduino)

Arduino memiliki open-source yang memudahkan untuk menulis kode dan meng-upload board ke arduino. Arduino *IDE* (*Integrated Development Enviroment*) ini merupakan media yang digunakan untuk memberikan informasi kepada arduino sehingga dapat memberikan output sesuai dengan apa yang diinginkan. Aplikasi arduino IDE ini dapat dijalankan di windows, Mac OS X, dan linux (Moh. Kamalul Wafi, 2014: 2). berikut merupakan gambaran tampilan arduino IDE :



```

Blink | Arduino 1.0
File Edit Sketch Tools Help
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);            // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}

```

**Gambar II.9. Tampilan Arduino IDE**

Sumber: [arduino.stackexchange.com](http://arduino.stackexchange.com) dikunjungi pada tanggal 29 Juli 2016

Dalam arduino terhubung dengan arduino *IDE* ini dengan hanya menekan tombol RESET. tombol ini dirancang untuk menjalankan program yang telah di upload ke arduino uno, tombol ini juga terhubung dengan ATMEga 328 melalui kapasitor 100nf. *IDE (Integrated Development Enviroment)* arduino merupakan pemograman dengan menggunakan bahasa C. Setiap program *IDE* arduino (biasa disebut *sketch*) mempunyai dua buah fungsi yang harus ada, yaitu :

a. `void setup() { }`

Semua kode didalam kurung kurawal akan dijalankan hanya satu kali ketika program *IDE* Arduino dijalankan untuk pertama kalinya.

b. *void loop()* { }

Fungsi ini akan dijalankan setelah setup (fungsi *void setup*) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya (power) dilepaskan.

*Compiler* merupakan modul yang berfungsi mengubah bahasa *processing* (kode pemrograman) kedalam kode biner karena kode biner adalah satu-satunya bahasa program yang dipahami oleh mikrokontroler. Sedangkan *upload* program adalah modul yang berfungsi memasukkan kode biner kedalam mikrokontroler. Pada *software* Arduino *IDE* memiliki fitur *compiler* sedangkan untuk *upload* program menggunakan *USBisp* yang dihubungkan ke port *ISP* pada papan rangkaian mikrokontroler. Pada proses ini akan merubah bahasa pemrograman dari digital ke bahasa analog yang dapat dipahami mikrokontroler (Anandya Bagus Venesa dan Wibowo Basuki Dwi, 2004 : 5).

### II.12.3. Android

Android adalah sistem operasi untuk perangkat selular yang berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi.

Android menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

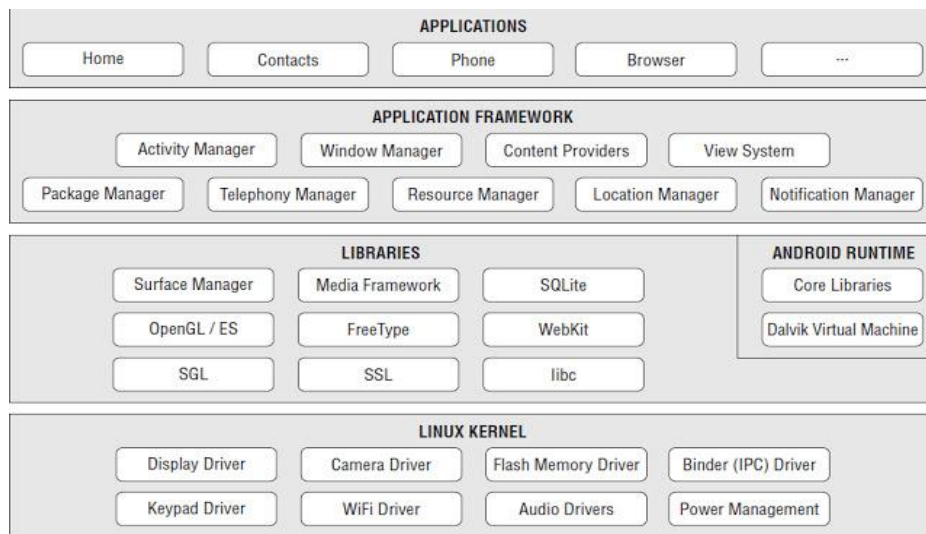
### II.12.3.1. Fitur – fitur Android

Android tersedia secara open source bagi manufaktur perangkat keras untuk modifikasinya sesuai kebutuhan. Meskipun konfigurasi perangkat Android tidak sama antara satu perangkat dengan perangkat lain. Fitur yang tersedia Android adalah :

1. Penyimpanan (*Storage*) : menggunakan SQLite yang merupakan database *relational* yang ringan untuk menyimpan data.
2. Koneksi (*Connectivity*) : mendukung *GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth* (termasuk *A2DP* dan *AVRCP*), *Wifi, LTE*, dan *WiMax*.
3. Pesan (*Messaging*) : mendukung SMS dan MMS.
4. Mendukung Media : audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
5. *Web Browser* : menggunakan *open source WebKit* termasuk di dalamnya *engine Chrome V8 JavaScript*.
6. *Hardware* : terdapat *Accelerometer Sensor, Camera, Digital Compass, Proximity Sensor* dan *GPS*.
7. *Multi touch* : mendukung layar *multi touch*.
8. *Multi tasking* : mendukung aplikasi *multi tasking*.

### II.12.3.2. Arsitektur Android

Agar lebih mudah memahami bagaimana Android bekerja, berikut ini bagan tingkatan – tingkatan sistem operasi Android :



**Gambar II.10. Arsitektur Sistem Operasi Android**

Sumber : *Pemograman Aplikasi Android*

Secara garis besar system operasi Android terbagi menjadi lima tingkatan :

1. **Applications** pada tingkat inilah kita akan bekerja, contoh aplikasi ini banyak ditemui, seperti : *Phone, Contack, Browse* dan lain – lain. Seperti aplikasi Android pada umumnya yang dapat di *download* dan di *install* dari Market Android. Semua aplikasi yang anda buat terleteak pada tingkat *Applications*.
2. **Application Framework** adalah semacam kumpulan class built-in yang tertanam dalam sistem operasi Android sehingga pengembang dapat memanfaatkannya untuk aplikasi yang sedang dibangun.
3. **Libraries** berisi semua kode program yang menyediakan layanan – layanan utama sistem operasi Android. Sebagai contoh library SQLite yang menyediakan dukungan database sehingga aplikasi Android dapat menggunakannya untuk menyimpan data. Library WebKit yang menyediakan fungsi – fungsi *browsing web*, dan lain – lain.

4. **Android Runtime** kedudukannya setingkat dengan libraries, Android *Runtime* menyediakan kumpulan pustaka inti yang dapat diaktifkan oleh pengembang untuk menulis kode aplikasi Android dengan bahasa pemrograman Java. *Dalvik Virtual Machine* aktif setiap kali aplikasi Android berproses (aplikasi Android dikompilasi menjadi *Dalvik executable*). *Dalvik* adalah mesin semu yang dirancang khusus untuk Android yang dapat mengoptimalkan daya *battery* perangkat bergerak dengan memori dan *CPU* terbatas.
5. **Linux kernel** adalah kernel dasar Android. Tingkat ini berisi semua *driver* perangkat tingkat rendah untuk komponen – komponen *hardware* perangkat Android.

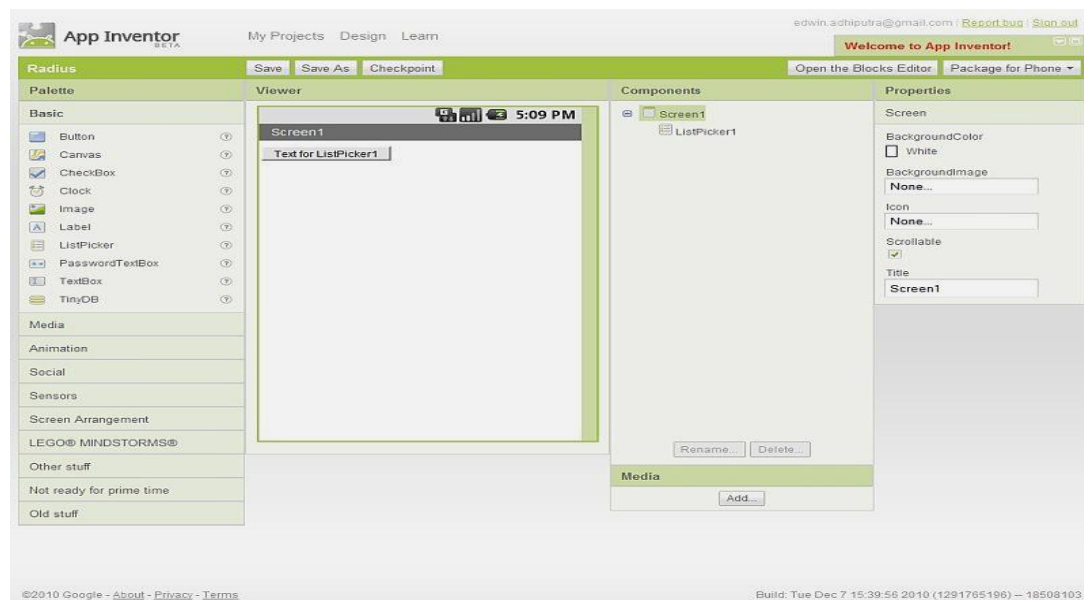
#### II.12.4. App Inventor

App Inventor for Android adalah aplikasi yang pada dasarnya disediakan oleh Google dan sekarang di-*maintenance* oleh Massachusetts Institute of Technology (MIT). App Inventor memungkinkan semua orang untuk membuat *software* aplikasi untuk sistem operasi android. Pengguna dapat menggunakan tampilan grafis GUI dan fitur drag and drop visual objek untuk membuat sebuah aplikasi dapat berjalan pada sistem operasi Android.

App Inventor adalah sebuah pemrograman visual yang digunakan untuk mengembangkan aplikasi berbasis Android dengan dukungan fitur berupa drag and drop tool. Anda dapat mendesain user *interface* dari sebuah aplikasi dengan menggunakan web GUI (*Graphical User Interface*) *builder*, kemudian anda dapat

menspesifikasikan behavior aplikasi dengan memasang *block* yang sesuai dengan kebutuhan anda.

App Inventor menggunakan *Kawa Language Framework* dan *Kawa's dialect* yang dikembangkan oleh Per Botner. Kedua aplikasi tersebut didistribusikan sebagai bagian dari GNU Operating System oleh *Free Software Foundation*. Kedua aplikasi tersebut dijadikan sebagai compiler dan menerjemahkan *Visual Block Programming* untuk di implementasikan pada *platform* Android.



**Gambar II.11. Tampilan Awal App Inventor**

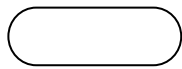





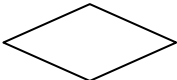


Sumber : Wahana Komputer : 2013 : 3

## II.13. Flowchart

Prinsip kerja dari pemodelan di atas dapat digambarkan melalui *flowchart*. *Flowchart* adalah gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta pernyataannya. Gambaran ini dinyatakan dengan simbol dan dengan demikian setiap simbol menggambarkan proses tertentu. Hubungan antar

proses digambarkan dengan garis penghubung (Zarlis et al, 2007). Flowchart disusun dengan simbol-simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Adapun tabel simbol *flowchart* pada Tabel II.1:

**Tabel II.1. Tabel Simbol Flowchart**

SIMBOL	NAMA	FUNGSI
	<b>TERMINATOR</b>	Permulaan/akhir program
	<b>GARIS ALIR (FLOW LINE)</b>	Arah aliran program
	<b>PREPARATION</b>	Proses inialisasi/pemberian harga awal
	<b>PROSES</b>	Proses perhitungan/proses pengolahan data
	<b>INPUT/OUTPUT DATA</b>	Proses input/output data, parameter, informasi
	<b>PREDEFINED PROCESS (SUB PROGRAM)</b>	Permulaan sub program/proses menjalankan sub program
	<b>DECISION</b>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	<b>ON PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	<b>OFF PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

Sumber : Dr. Suarga, M. sc., M. Math. , Ph D. , 2012.



### **BAB III**

#### **ANALISA DAN PERANCANGAN**

## **BAB III**

### **ANALISA DAN PERANCANGAN**

#### **III.1. Analisa Masalah**

Dalam perancangan robot penyeimbang menggunakan sensor jarak berbasis android, terdapat beberapa masalah yang harus dipecahkan.

Permasalahan tersebut antara lain :

##### 1. Keseimbangan Robot

Masalah awal dan yang paling utama dalam perancangan robot keseimbangan adalah bagaimana membuat robot dapat menyeimbangkan dirinya, dan mampu mempertahankan posisinya ketika dalam keadaan seimbang. Karena dibutuhkan ketepatan sensor dalam menentukan sudut kemiringan badan robot dan keluaran tegangan baterai dalam mengatur kecepatan putaran roda agar tidak terjadi salah perhitungan yang menyebabkan robot terjatuh. Ketika sensor telah membaca sudut kemiringan badan robot, maka informasi tersebut dikirimkan ke mikrokontroler agar dapat mengatur kecepatan perputaran roda sehingga robot dapat mempertahankan keseimbangannya.

##### 2. Penggunaan Komunikasi *Bluetooth*

Masalah yang kedua dalam perancangan robot keseimbangan adalah penggunaan *bluetooth* yang merupakan media untuk memberi perintah kepada robot agar dapat maju, mundur dan berbelok. *Bluetooth* akan bekerja sesuai perintah yang dikirimkan oleh *smartphone*. *Bluetooth* sendiri

terkoneksi dengan *smartphone android* dan yang terpasang di robot keseimbangan.

### 3. Sensor Jarak Sebagai Penghindar Objek

Perancangan robot penyeimbang menggunakan sensor jarak berbasis android ini menggunakan tambahan sensor untuk mendeteksi objek yang ada pada bagian depan robot. Sensor jarak ini dimaksudkan untuk mencegah robot mengalami kerusakan ketika menabrak dinding secara langsung.

## III.2. Strategi Pemecahan masalah

Ada beberapa permasalahan yang terjadi dalam perancangan robot penyeimbang menggunakan sensor jarak berbasis android, untuk itu dibutuhkan solusi atau pemecahan masalah, antara lain:

1. Dengan adanya permasalahan pada robot keseimbangan, penulis harus lebih teliti dalam menentukan sudut kemiringan pada robot dan keluaran tegangan untuk motor agar berfungsi maksimal sesuai dengan kebutuhan pada sistem yang bekerja. Dalam hal ini penulis menggunakan PID sebagai pengontrol kecepatan robot dan mengatur pergerakan pada robot.
2. Untuk permasalahan *bluetooth*, penulis menggunakan *bluetooth* tipe HC-05 sebagai perangkat yang menerima informasi dari *smartphone user* (pengguna), *user* sendiri dapat mengirim perintah melalui aplikasi yang telah dirancang sebelumnya.

3. Sensor jarak yang digunakan adalah Sharpgp GP12. Sensor ini menggunakan cahaya infra merah yang kasat mata (tidak terlihat) dan data yang cukup akurat dalam mendeteksi objek.

### **III.3. Identifikasi Kebutuhan**

Adapun identifikasi kebutuhan dari perancangan robot penyeimbang menggunakan sensor jarak berbasis android yaitu analisis kebutuhan *hardware*, dan analisis kebutuhan *software*.

#### **III.3.1. Kebutuhan Perangkat Keras (*Hardware*) Perancangan *Interface***

Dalam perancangan robot penyeimbang menggunakan sensor jarak berbasis android ini menggunakan perangkat keras (*hardware*) dengan spesifikasi sebagai berikut :

1. *Core i3; Processor 2.53GHz*
2. *Hard disk : 320 GB*
3. *RAM 2 GB*
4. *Laptop*
5. *Keyboard dan Mouse.*
6. *Smartphone Android*

#### **III.3.2. Kebutuhan Perangkat Keras (*Hardware*) Perancangan Alat**

Adapun kebutuhan perangkat lain dalam perancangan alat adalah sebagai berikut:

1. ArduinoUno R3with ATMega328.
2. Sensor MPU6050.
3. *Bluetooth HC – 05*.
4. *Sharpgp GP12*.
5. *Driver Motor L298*.
6. Baterai Lipo 3 cell 11.1 Volt 1 Ampere.
7. Motor DC 400 rpm.
8. Komponenpendukunglainnya.

### **III.3.3.Kebutuhan Perangkat Lunak (*Software*)**

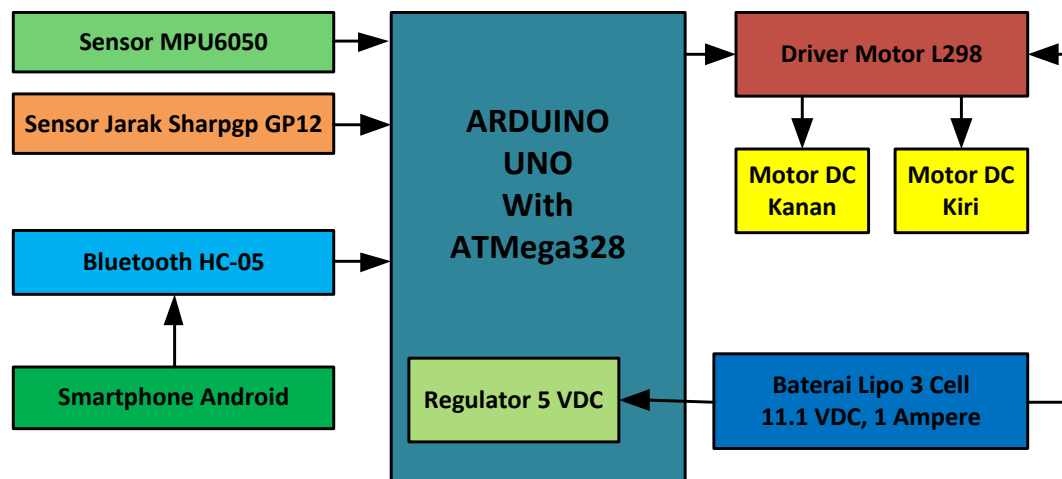
Adapun perangkat lunak (*software*) yang digunakan perancangan robot penyeimbangmenggunakan sensor jarakberbasis androidini adalah sebagai berikut :

1. Sistem operasi *Windows 7 32bit*.
2. Arduino IDE 1.6.5
3. *App Inventor 2 Ultimate*
4. Sistem Operasi Android min. *Ice Cream Sandwich*

### **III.4. Diagram Blok Rangkaian**

Secara garis besar, perancangan robot penyeimbang menggunakan sensor jarak berbasis android ini terdiri dari Arduino Uno R3with ATMega328, Sensor MPU6050, *Bluetooth HC – 05*, Sensor Jarak *Sharpgp GP12*, *Driver Motor L298*, BateraiLipo 3 cell 11.1 Volt 1 Ampere, Motor DC 400 rpm, serta *smartphone*

android. Adapun diagram blok dari sistem yang dirancang adalah seperti yang diperlihatkan pada gambar III.1. berikut ini:



**Gambar III.1. Diagram Blok Rangkaian**

Penjelasan dan fungsi dari masing – masing blok adalah sebagai berikut:

- Smartphone android* berfungsi untuk memberi perintah maju, mundur atau berbelok.
- Bluetooth HC-05* berfungsi sebagai pengirim data dari *smartphone android* ke mikrokontroler *arduinouno*.
- Power supply* berfungsi sebagai sumber energy atau tegangan, berasal dari baterai lipo 3 sel dengan tegangan 11,1 Volt 1 Ampere.
- Arduinouno* dengan mikrokontroler ATmega328 berfungsi sebagai pusat kendali dari keseluruhan system kerja rangkaian.
- MPU6050 sebagai sensor *accelerometer* dan *Gyroscope* yang sudah terintegrasi.
- Sensor sharp GP2D12 digunakan untuk membaca jarak dinding / objek.
- Regulator tegangan 5 VDC, terintegrasi dengan rangkaian *arduino uno*.

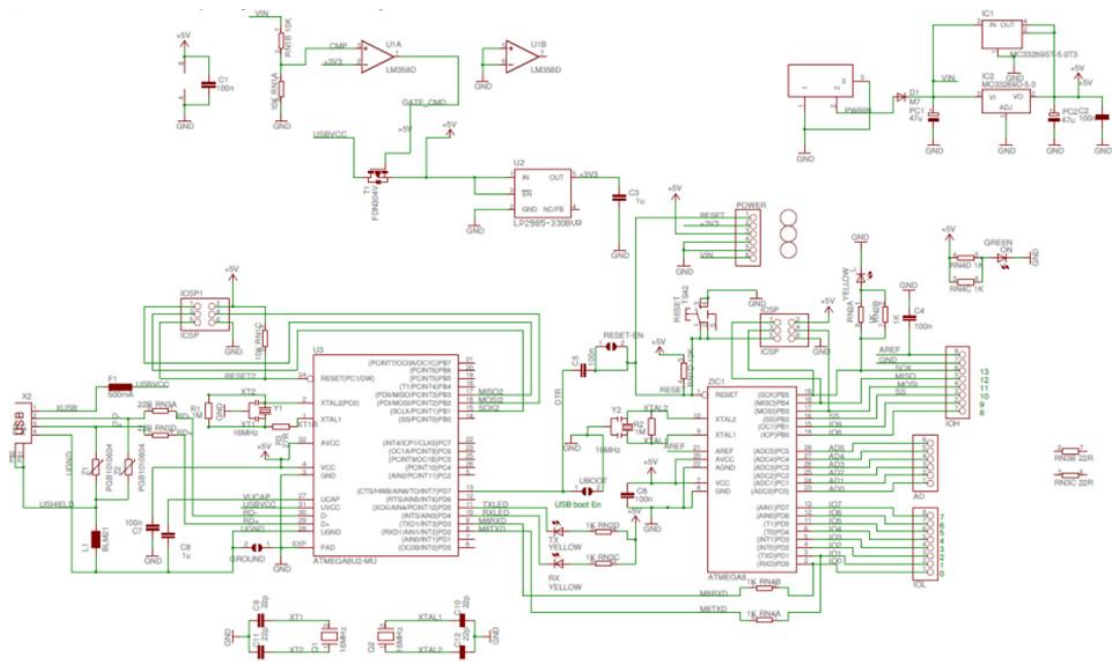
- h. *Motor Driver*L298 berfungsi sebagai pengatur kecepatan pada motor.
- i. *Motor DC* pada bagian kanan dan kiri berfungsi sebagai penggerak robot.

### **III.5. Perancangan**

#### **III.5.1. Perancangan Elektronik**

##### **III.5.1.1. Sistem Minimum Rangkaian Mikrokontroler ATmega 328**

Arduino adalah kit elektronik atau papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan ATMEL. Komponen utama dari rangkaian *Arduinouno* adalah IC Mikrokontroler sebagai prosesnya. Mikrokontroler yang digunakan dalam system inia adalah mikrokontroler dengan jenis AVR seri ATmega328. Mikrokontroler ini mempunyai 20 pin yang meliputi 14 pin *I/O digital* dengan 6 pin yang dapat berfungsi sebagai *output PWM (Pulse Width Module)* dan 6 pin *I/O analog*. Pemilihan ATmega328 ini dikira akan memaksimalkan pembuatan alat sebagai pengolah data. Rangkaian mikrokontroler ATmega328 pada *Arduino Uno* dapat dilihat pada gambar III.2.



**Gambar III.2 SkematikRangkaianMikrokontroler ATmega328PadaArduinoUno**

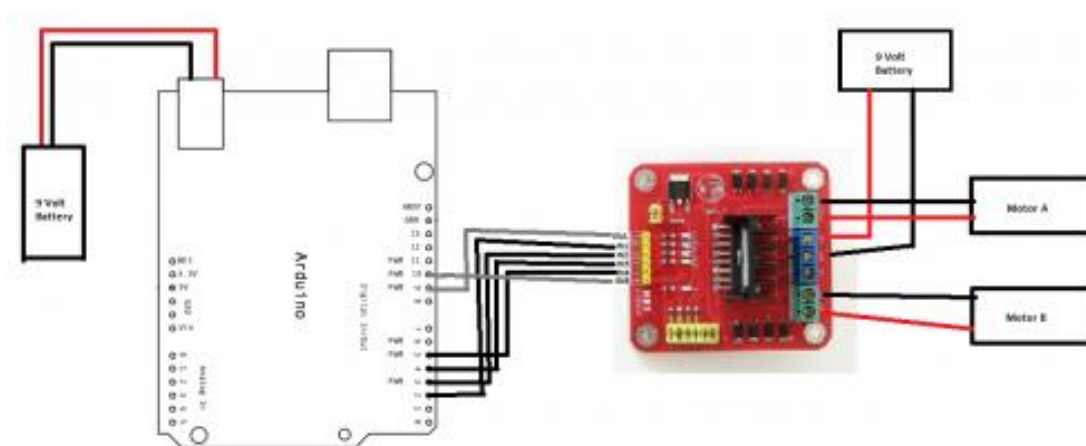
Berikut adalah tabel penggunaan *port input/output* pada perancangan robot penyeimbang menggunakan sensor jarakberbasis android.

**Tabel III.1Konfigurasi Pin PadaArduino**

Nama PIN/PORT Arduino	Fungsi	Keterangan
A1	Input	Sensor Jarak SharpGP12
A2	Input	Sensor Jarak SharpGP12
A4	Input	Data SDA MPU6050
A5	Input	Data SCL MPU6050
D0	Input	RX HC-05 <i>Bluetooth</i>
D1	Output	TX HC-05 <i>Bluetooth</i>
D2	Output	Data INT MPU6050
D4	Output	Driver Motor L298
D5	Output	Driver Motor L298
D6	Output	Driver Motor L298
D7	Output	Driver Motor L298

### III.5.1.2. Rangkaian Driver Motor L298

Rangkaian *driver motor L298* ini berfungsi sebagai pengatur kecepatan motor. Rangkaian *driver motor* dapat dilihat pada gambar berikut:



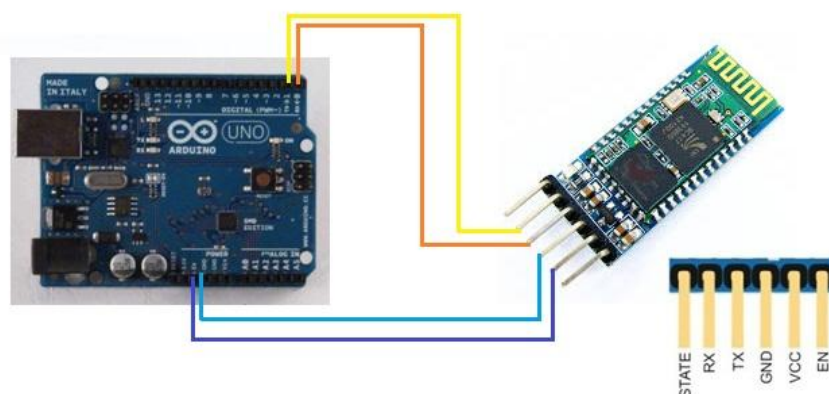
**Gambar III.3** Skematik Rangkaian Driver Motor L298

Pada gambar III.3, VCC pada driver motor L298 dihubungkan ke kutub positif baterai dan kaki negatif driver motor dihubungkan ke kutub negatif baterai. *Forward* dan *reverse* dihubungkan ke mikrokontroler. Jika *forward* bernilai 1 maka motor kaki motor A akan terhubung dengan kutub positif dan B terhubung dengan kutub negatif yang mengakibatkan arah perputaran motor searah dengan jarum jam. Sebaliknya jika *reverse* bernilai 1 maka motor kaki motor B akan terhubung dengan kutub positif dan A terhubung dengan kutub negatif yang mengakibatkan motor berputar berlawanan dengan arah jarum jam.

### III.5.1.3. Rangkaian Bluetooth HC-05

*Bluetooth HC-05* ini berfungsi sebagai media pengiriman data dari mikrokontroler ke smartphone. *Bluetooth* ini berperan penting di dalam sistem karena *Bluetooth* ini berperan sebagai pengiriman data. Kaki Vcc pada *Bluetooth*

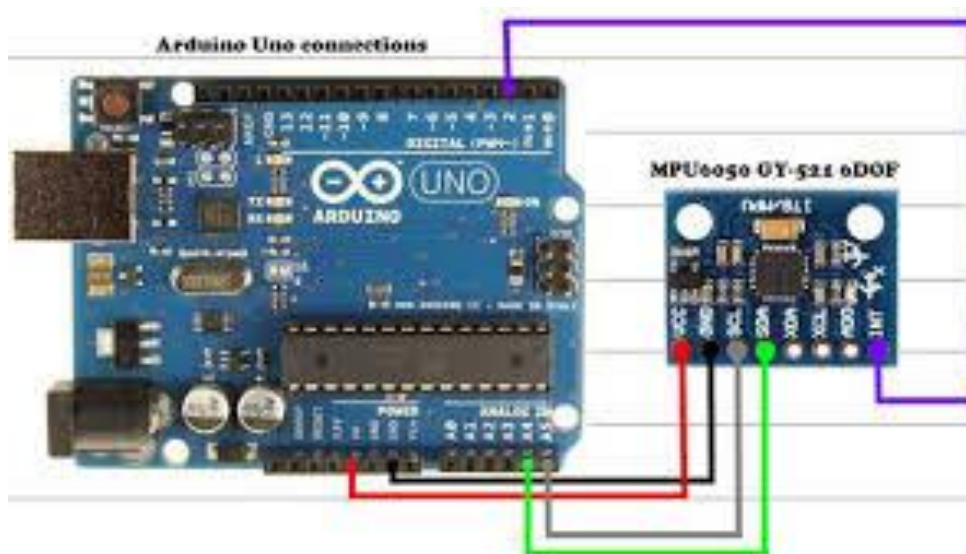
dihubungkan ke VCC *arduinouno* dan kaki GND dihubungkan ke GND mikrokontroler. Kaki TX dihubungkan dengan RX pada arduino sedangkan kaki RX dihubungkan dengan kaki TX pada *arduino uno*. Skematik *Bluetooth* pada arduino dapat dilihat dari Gambar III.4 berikut:



**Gambar III.4 Skematik Pemasangan *Bluetooth* HC – 05 Pada *Arduino* Uno**

#### III.5.1.4. Rangkaian Sensor MPU6050

MPU6050 adalah *chip IC invense* yang di dalamnya terdapat sensor *accelerometer* dan *Gyroscope* yang sudah terintegrasi. Alasan menggunakan sensor ini adalah karena harganya relative murah dimana sudah mendapatkan 2 sensor yang sudah terintegrasi. MPU6050 berfungsi sebagai keseimbangan pada robot. Sensor ini memberikan data sudut kemiringan badan robot kepada mikrokontroler, sehingga mikrokontroler dapat memproses data tersebut dan membuat badan robot tetap seimbang. Skematik sensor MPU6050 dapat dilihat pada gambar berikut :

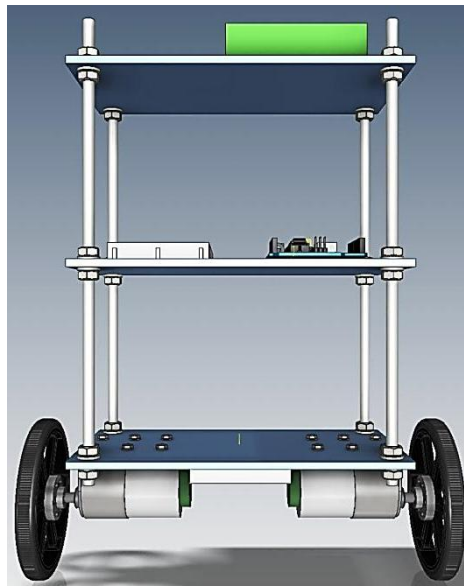


**Gambar III.5** Skematik Pemasangan MPU6050 Pada Arduino Uno

Pada gambar III.5, kaki Vcc pada MPU6050 dihubungkan ke VCC *arduinouno*, kaki GND dihubungkan ke GND mikrokontroler. Kaki SDA dihubungkan dengan A5 pada *arduino uno* sedangkan kaki SCL dihubungkan dengan kaki A4 pada arduino dan kaki int dihubungkan ke pin 2 arduino.

### III.5.2. Perancangan Mekanik (*Hardware*)

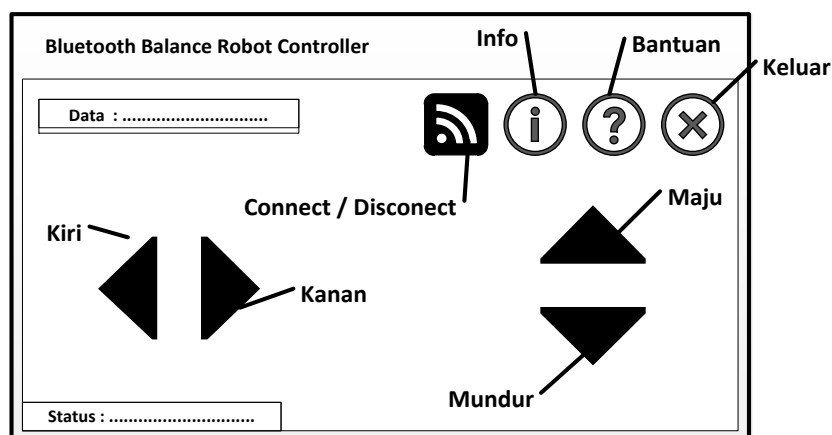
Perancangan mekanik dari robot penyeimbang menggunakan sensor jarak berbasis android menggunakan bentuk kotak (*box*) berbahan plastik atau akrilik dengan ketebalan 3mm. Dimensi robot yang digunakan dengan panjang sebesar 25 cm, lebar sebesar 15 cm dan tinggi sebesar 20 cm Berikut adalah gambar dari perancangan mekanik tersebut.



**Gambar III.6 Perancangan Mekanik**

### **III.5.3. Perancangan Perangkat Lunak (*Software*)**

Perancangan *software* pada robot penyeimbang menggunakan sensor jarak berbasis android dimulai mendesain tampilan *form* untuk alat yang akan dibuat. Setelah itu akan dirancang pembuatan program untuk alat yang akan dibuat. Perancangan tampilan pada program *app inventor 2 ultimate* bertujuan untuk menggambarkan sketsa desain tampilan program yang akan dibuat sebagai *interface* kepada pengguna aplikasi. Berikut rancangan tampilan *form* untuk perancangan robot penyeimbang menggunakan sensor jarak berbasis android:



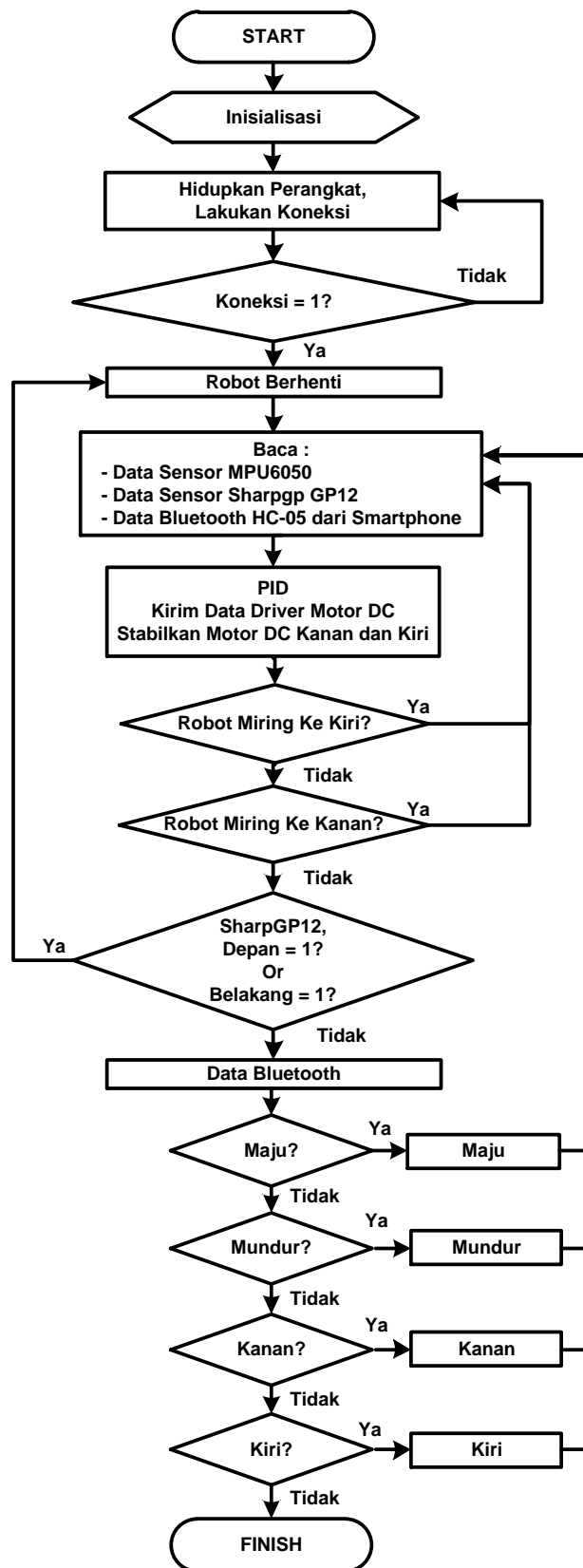
**Gambar III.7. Desain Tampilan pada Smartphone Android**

Pada gambar III.7, tampilan yang akan dibuat dapat menampilkan beberapa perintah untuk robot keseimbangan yaitu maju, mundur, belok kiri dan belok kanan. Dalam perancangan aplikasi terdapat menu koneksi dan tombol diskoneksi, menu info yang menerangkan tentang pembuat, menu bantuan dan menu keluar. Pada menu info dijelaskan fungsi dari bagian-bagian dari aplikasi.

### III.6. Flowchart

#### III.6.1. Flowchart Perangkat Robot Keseimbangan

Agar dapat melihat struktur jalannya program maka dibuat *flowchart* (diagram alur). *Flowchart* digunakan sebagai dasar acuan dalam membuat program. Struktur program akan lebih mudah dibuat atau didesain. Selain itu juga jika terdapat kesalahan akan lebih mudah untuk mendeteksi letak kesalahannya serta untuk lebih memudahkan dalam menambahkan instruksi-instruksi baru pada program jika nantinya terjadi pengembangan pada struktur programnya. Adapun *flowchart* dari sistem yang dirancang adalah seperti yang diperlihatkan pada gambar III.8:

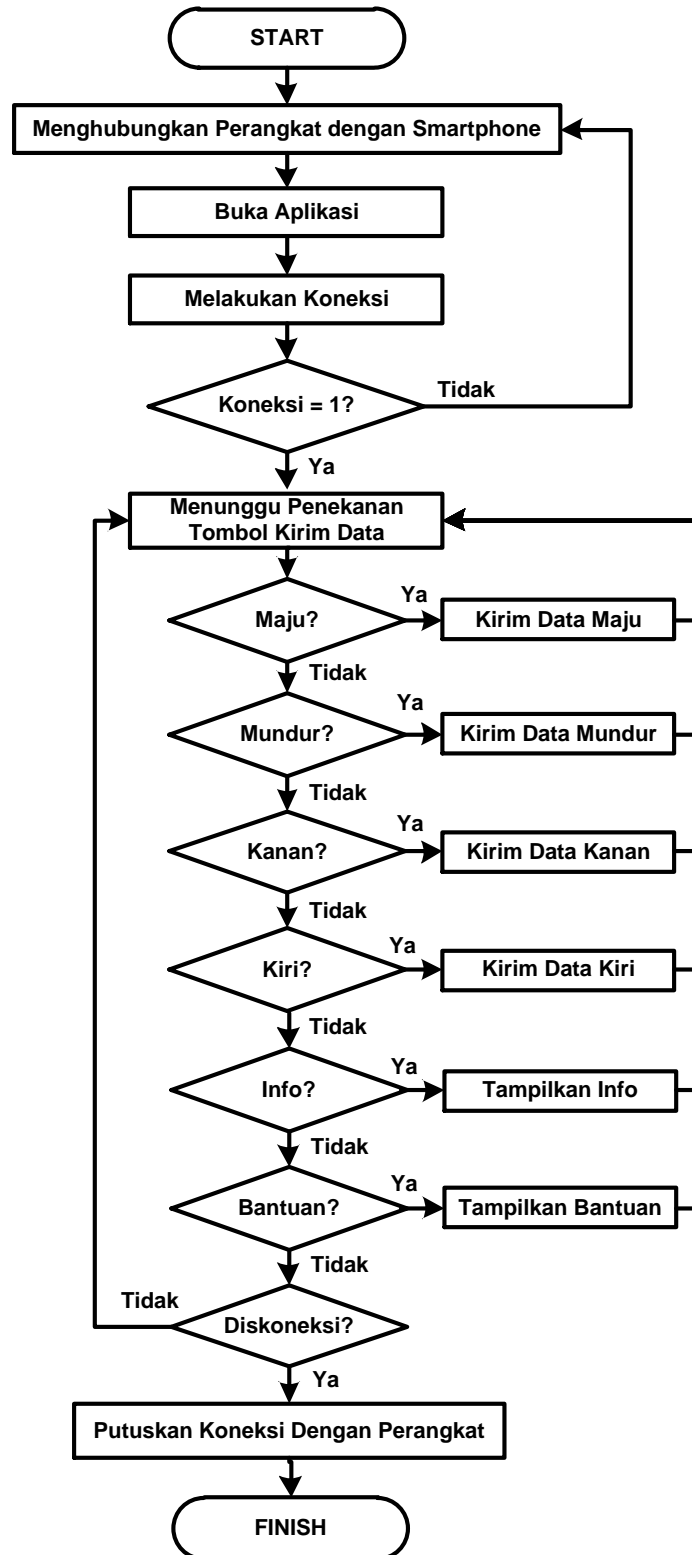


Gambar III.8. Flowchart Perangkat Robot

Penjelasan *algoritma* perangkat robot keseimbangan:

1. *Start*
2. Hidupkan perangkat dengan menghubungkan tegangan pada rangkaian.
3. Inisialisasi MPU6050 dimaksudkan untuk membaca sudut kemiringan robot.
4. Perangkat menerima data sensor MPU6050, data sensor SharpGP GP12 dan data dari aplikasi *smartphone* android melalui *bluetooth*.
5. Jika sudah seimbang maka robot akan berhenti dan melakukan pengereman untuk mempertahankan keseimbangan robot dan pembacaan sensor akan kembali dilakukan. Jika keadaan robot belum seimbang maka akan dibandingkan sudut kemiringannya
6. Jika robot miring kedepan maka robot bergerak maju dan sebaliknya.
7. Kontrol PID akan mengatur berapa tegangan yang akan dikeluarkan *driver* motor DC dan diteruskan ke motor DC.
8. Tegangan yang diperoleh dari perhitungan PID lalu diteruskan ke PWM agar percepatan motor menjadi lebih baik.
9. Kembali lagi dalam proses pembacaan sensor.
10. Jika sensor jarak depan atau sensor jarak belakang menerima data, robot akan berhenti dan menghindari objek.
11. Jika perangkat menerima data dari aplikasi *smartphone* android melalui *bluetooth* berupa maju, maka robot akan bergerak maju. Sama halnya dengan perintah mundur, kanan dan kiri.
12. *Finish*.

### III.6.2. Flowchart Interface



Gambar III.9. Flowchart Interface

Penjelasan algoritma *flowchart interface* :

1. Start
2. Menghubungkan perangkat mikrokontroler dengan *smartphone*.
3. Membuka aplikasi yang telah dirancang.
4. Melakukan koneksi, jika sudah terkoneksi maka akan dapat mengirim perintah untuk robot keseimbangan pada layar *smartphone*.
5. Jika tombol yang ditekan adalah maju, maka aplikasi akan mengirimkan data maju ke perangkat robot. Sama halnya dengan tombol mundur, kanan dan kiri.
6. Jika tombol info ditekan, maka aplikasi akan menampilkan info pembuat.
7. Jika tombol bantuan ditekan, maka aplikasi akan menampilkan bantuan penggunaan aplikasi.
8. Jika tombol diskoneksi ditekan, maka sambungan koneksi *bluetoothsmartphone* android dengan perangkat robot akan terputus.
9. Jika tidak ada tombol yang ditekan, maka aplikasi akan tetap menunggu data dari pengguna.
10. Finish.



## **BAB IV**

### **HASIL DAN UJI COBA**

## **BAB IV**

### **HASIL DAN UJI COBA**

#### **IV.1. Tampilan Hasil**

Dalam bab ini akan dijelaskan dan ditampilkan bagaimana hasil dari pengujian rancangan alat yang dibuat beserta pembahasan tentang pergerakan dan cara kerja perangkat robot penyeimbang menggunakan sensor jarak berbasis android. Adapun hasil dari pengujian yang dilakukan adalah perangkat elektronik yang dibuat atau dirancang dan diprogram dengan menggunakan aplikasi Arduino IDE.

#### **IV.2. Pelaksanaan Pengujian Rangkaian**

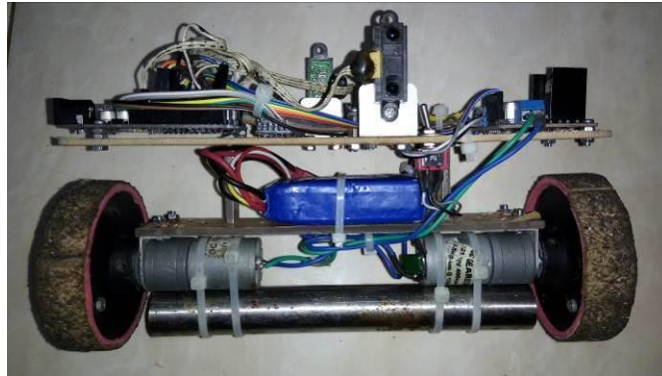
Sebelum melakukan pengujian, beberapa hal yang harus diperhatikan dan dipersiapkan adalah sebagai berikut :

1. Perangkat robot penyeimbang menggunakan sensor jarak dalam keadaan siap diuji, tidak ada *trouble* pada saat pengujian.
2. Sebelum pengujian perangkat, hubungkan *power* (baterai lipo) dengan perangkat dan menhidupkan tombol *power on/off*.
3. Hasil pengujian dianalisa dan dibandingkan dengan perangkat pembanding, seperti menghitung jarak sensor dengan penggaris, menghitung tegangan menggunakan multimeter dan lain sebagainya.
4. Hasil pengujian dipaparkan dalam bentuk tabel dan grafik, dianalisa dan dijelaskan secara terperinci.

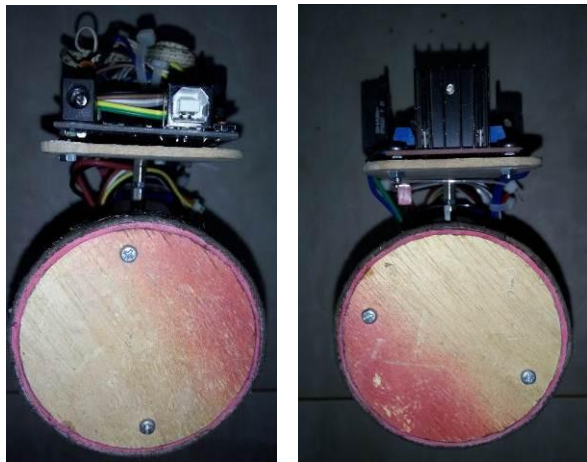
### IV.3. Tampilan Hasil Perangkat

Berikut adalah tampilan hasil perancangan robot penyeimbang menggunakan sensor jarak berbasis android, ditunjukkan oleh gambar di bawah ini:

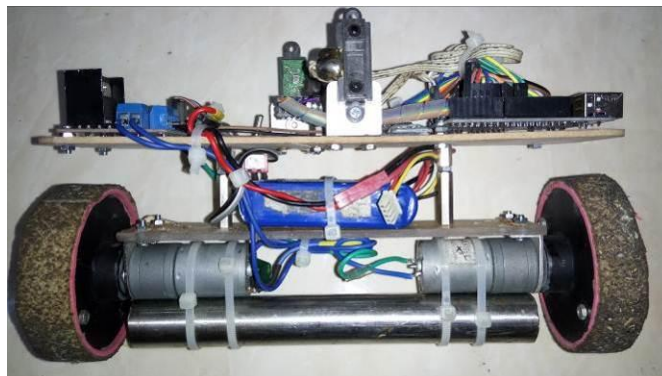
#### A. Sisi Depan



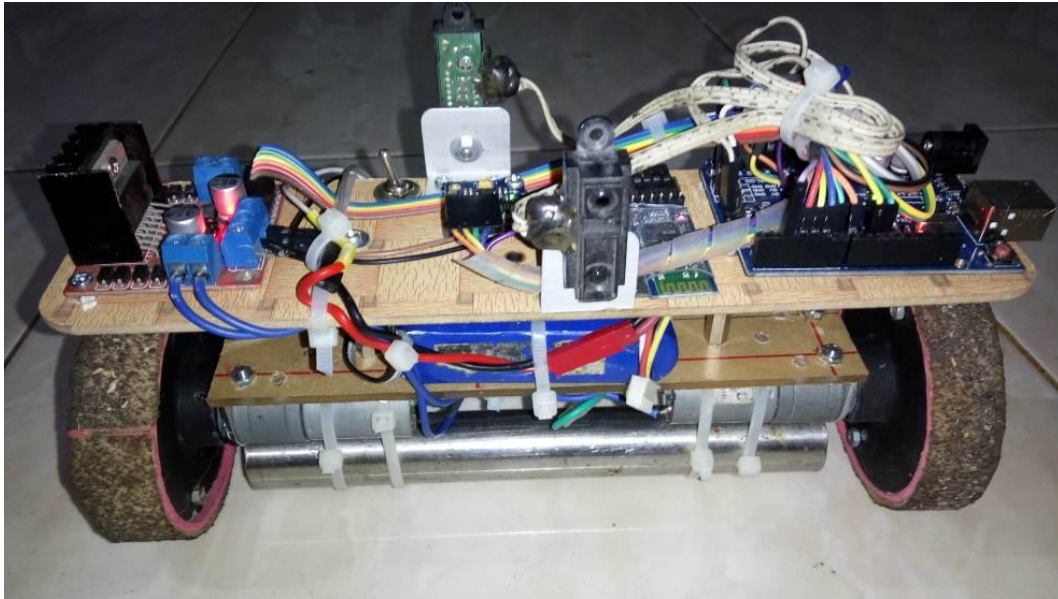
#### B. Sisi Samping



#### C. Sisi Belakang



#### D. Hasil Keseluruhan




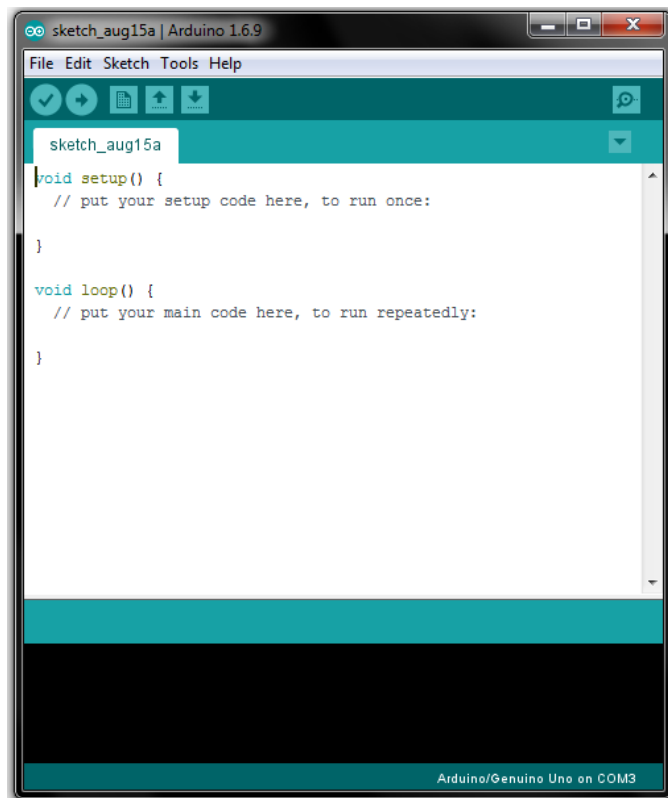
**Gambar IV.1. Perangkat Keseluruhan**

#### IV.4. Pengujian *Software*

Untuk mengetahui apakah rangkaian pada perangkat telah bekerja dengan baik, maka dilakukan pengujian dengan memberikan program perintah pada mikrokontroler dengan melakukan penginputan data dari komputer ke dalam mikrokontroler. Sebelum dilakukannya proses *download* program, hubungkan terlebih dahulu antara komputer melalui kabel USB dengan rangkaian mikrokontroler.

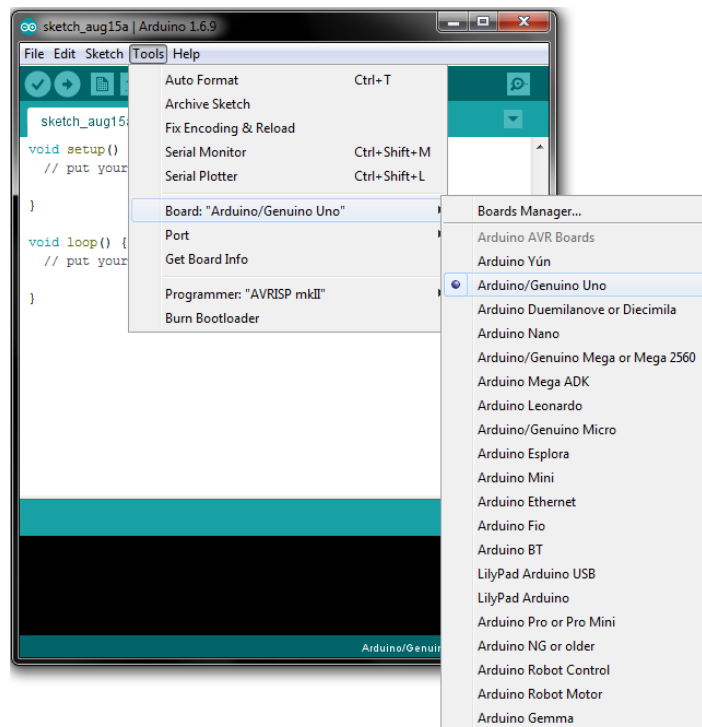
Dalam proses instalasi ini menggunakan aplikasi *Arduino 1.6.9*. Untuk melakukan instalasi ini dapat dilakukan dengan beberapa langkah antara lain :

- a. Langkah pertama yang dilakukan adalah dengan mengklik *icon*  *Arduino*. Setelah program melakukan *load* maka akan terlihat bentuk tampilan seperti gambar IV.2 berikut.

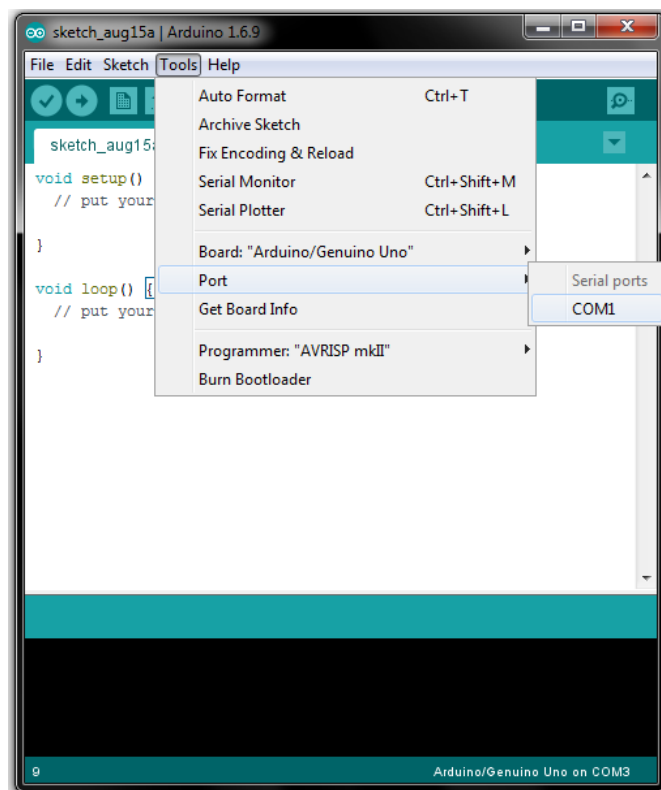


**Gambar IV.2. Tampilan Arduino 1.6.9**

- b. Selanjutnya yang dilakukan sebelum menginstal program terhadap mikrokontroler adalah melakukan pengaturan (*setting*) pada perangkat yang diperlukan dan menyetikkan program sesuai dengan yang dibutuhkan. Pengaturan pertama adalah pemilihan *board* arduino yang digunakan pada *software* sesuai dengan perangkat yaitu Arduino UNO, seperti pada gambar IV.3. Pengaturan kedua adalah pemilihan *port USB* yang digunakan perangkat, seperti pada gambar IV.4. berikut :



**Gambar IV.3. Pengaturan dan Pemilihan *Board* Arduino**



**Gambar IV.4. Pengaturan *Port* USB pada *Software* Arduino 1.6.9**

- c. Setelah pengaturan selesai, proses berikutnya adalah penulisan *listing* program. Berikut adalah *listing* program dari perangkat robot penyeimbang menggunakan sensor jarak berbasis android:

```

//*****
#include <PID_v1.h>
#include <LMotorController.h>
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    #include "Wire.h"
#endif

#define LOG_INPUT 0
#define MANUAL_TUNING 0
#define LOG_PID_CONSTANTS 0 //MANUAL_TUNING must be 1
#define MOVE_BACK_FORTH 0
#define MIN_ABS_SPEED 30

#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11); // RX, TX

//MPU
MPU6050 mpu;

// MPU control/status vars
bool dmpReady = false; // set true if DMP init was successful
uint8_t mpuIntStatus; // holds actual interrupt status byte from MPU
uint8_t devStatus; // return status after each device operation (0 = success, != 0 = error)
uint16_t packetSize; // expected DMP packet size (default is 42 bytes)
uint16_t fifoCount; // count of all bytes currently in FIFO
uint8_t fifoBuffer[64]; // FIFO storage buffer

// orientation/motion vars
Quaternion q; // [w, x, y, z] quaternion container
VectorFloat gravity; // [x, y, z] gravity vector
float ypr[3]; // [yaw, pitch, roll] yaw/pitch/roll container and gravity vector

//PID
#if MANUAL_TUNING
    double kp , ki, kd;
    double prevKp, prevKi, prevKd;

```

```

#endif
double originalSetpoint = 172.2;
double setpoint = originalSetpoint;
double movingAngleOffset = 1;
double input, output;
int moveState=0; //0 = balance; 1 = back; 2 = forth
char perintah;

#if MANUAL_TUNING
  PID pid(&input, &output, &setpoint, 0, 0, 0, DIRECT);
#else
  PID pid(&input, &output, &setpoint, 20, 80, 1.3, DIRECT);
#endif

//MOTOR CONTROLLER
int ENA = 3;
int IN1 = 4;
int IN2 = 8;
int IN3 = 5;
int IN4 = 7;
int ENB = 6;
LMotorController motorController(ENA, IN1, IN2, ENB, IN3, IN4, 0.6, 1);

//timers
long time1Hz = 0;
long time5Hz = 0;

volatile bool mpuInterrupt = false; // indicates whether MPU interrupt pin has
gone high
void dmpDataReady()
{
  mpuInterrupt = true;
}

void setup()
{
  //motorController.move(150,150,10);
  //delay(1000);
  //motorController.move(0,0,10);
  ///delay(1000);
  //motorController.move(-150,-150,10);
  ///delay(1000);
  //motorController.move(0,0,10);
  //delay(1000);
  // join I2C bus (I2Cdev library doesn't do this automatically)

```

```

#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
  Wire.begin();
  TWBR = 24; // 400kHz I2C clock (200kHz if CPU is 8MHz)
#elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
  Fastwire::setup(400, true);
#endif

// initialize serial communication
// (115200 chosen because it is required for Teapot Demo output, but it's
// really up to you depending on your project)
Serial.begin(115200);
mySerial.begin(9600);
while (!Serial); // wait for Leonardo enumeration, others continue immediately

// initialize device
Serial.println(F("Initializing I2C devices..."));
mpu.initialize();

// verify connection
Serial.println(F("Testing device connections..."));
Serial.println(mpu.testConnection() ? F("MPU6050 connection successful") :
F("MPU6050 connection failed"));

// load and configure the DMP
Serial.println(F("Initializing DMP..."));
devStatus = mpu.dmpInitialize();

// supply your own gyro offsets here, scaled for min sensitivity
mpu.setXGyroOffset(682);
mpu.setYGyroOffset(524);
mpu.setZGyroOffset(-271);
mpu.setZAccelOffset(25816); // 1688 factory default for my test chip

// make sure it worked (returns 0 if so)
if (devStatus == 0)
{
  // turn on the DMP, now that it's ready
  Serial.println(F("Enabling DMP..."));
  mpu.setDMPEnabled(true);

  // enable Arduino interrupt detection
  Serial.println(F("Enabling interrupt detection (Arduino external interrupt
0)..."));
  attachInterrupt(0, dmpDataReady, RISING);
  mpuIntStatus = mpu.getIntStatus();

```

```

// set our DMP Ready flag so the main loop() function knows it's okay to use
it
Serial.println(F("DMP ready! Waiting for first interrupt..."));
dmpReady = true;

// get expected DMP packet size for later comparison
packetSize = mpu.dmpGetFIFOpacketSize();

//setup PID
pid.SetMode(AUTOMATIC);
pid.SetSampleTime(10);
pid.SetOutputLimits(-255, 255);
}
else
{
  // ERROR!
  // 1 = initial memory load failed
  // 2 = DMP configuration updates failed
  // (if it's going to break, usually the code will be 1)
  Serial.print(F("DMP Initialization failed (code "));
  Serial.print(devStatus);
  Serial.println(F(")"));
}
}

void loop()
{
  // if programming failed, don't try to do anything
  if (!dmpReady) return;

  // wait for MPU interrupt or extra packet(s) available
  while (!mpuInterrupt && fifoCount < packetSize)
  {
    //no mpu data - performing PID calculations and output to motors
    pid.Compute();
    motorController.move(output, MIN_ABS_SPEED);

    unsigned long currentMillis = millis();
    if (currentMillis - time1Hz >= 1000)
    {
      loopAt1Hz();
      time1Hz = currentMillis;
    }

    if (currentMillis - time5Hz >= 5000)
    {

```

```

        loopAt5Hz();
        time5Hz = currentMillis;
    }
}

// reset interrupt flag and get INT_STATUS byte
mpuInterrupt = false;
mpuIntStatus = mpu.getIntStatus();

// get current FIFO count
fifoCount = mpu.getFIFOCount();

// check for overflow (this should never happen unless our code is too
inefficient)
if ((mpuIntStatus & 0x10) || fifoCount == 1024)
{
    // reset so we can continue cleanly
    mpu.resetFIFO();
    Serial.println(F("FIFO overflow!"));

// otherwise, check for DMP data ready interrupt (this should happen
frequently)
}
else if (mpuIntStatus & 0x02)
{
    // wait for correct available data length, should be a VERY short wait
    while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount();

    // read a packet from FIFO
    mpu.getFIFOBytes(fifoBuffer, packetSize);

    // track FIFO count here in case there is > 1 packet available
    // (this lets us immediately read more without waiting for an interrupt)
    fifoCount -= packetSize;

    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
    #if LOG_INPUT
        Serial.print("ypr\t");
        Serial.print(ypr[0] * 180/M_PI);
        Serial.print("\t");
        Serial.print(ypr[1] * 180/M_PI);
        Serial.print("\t");
        Serial.println(ypr[2] * 180/M_PI);
    #endif
}

```

```

    input = ypr[1] * 180/M_PI + 180;

    if (mySerial.available()) {
    perintah = mySerial.read();
    if(perintah=='F'){
    Serial.println("F");
    motorController.move(150,150,10); delay(50);
    }
    else if(perintah=='B'){
    Serial.println("B");
    motorController.move(-150,-150,10); delay(50);
    }
    else if(perintah=='L'){
    Serial.println("L");
    motorController.move(-150,150,10); delay(50);
    }
    else if(perintah=='R'){
    Serial.println("R");
    motorController.move(150,-150,10); delay(50);
    }
    else if(perintah=='S'){
    Serial.println("S");
    mpu.resetFIFO();
    }
    }
}

void loopAt1Hz()
{
#if MANUAL_TUNING
    setPIDTuningValues();
#endif
}

void loopAt5Hz()
{
    #if MOVE_BACK_FORTH
        moveBackForth();
    #endif
}

//move back and forth
void moveBackForth()
{
    moveState++;

```

```

if (moveState > 2) moveState = 0;
if (moveState == 0)
  setpoint = originalSetpoint;
else if (moveState == 1)
  setpoint = originalSetpoint - movingAngleOffset;
else
  setpoint = originalSetpoint + movingAngleOffset;
}

//PID Tuning (3 potentiometers)
#if MANUAL_TUNING
void setPIDTuningValues()
{
  readPIDTuningValues();
  if (kp != prevKp || ki != prevKi || kd != prevKd)
  {
    #if LOG_PID_CONSTANTS
      Serial.print(kp);Serial.print(",          ");Serial.print(ki);Serial.print(",
");Serial.println(kd);
    #endif
    pid.SetTunings(kp, ki, kd);
    prevKp = kp; prevKi = ki; prevKd = kd;
  }
}

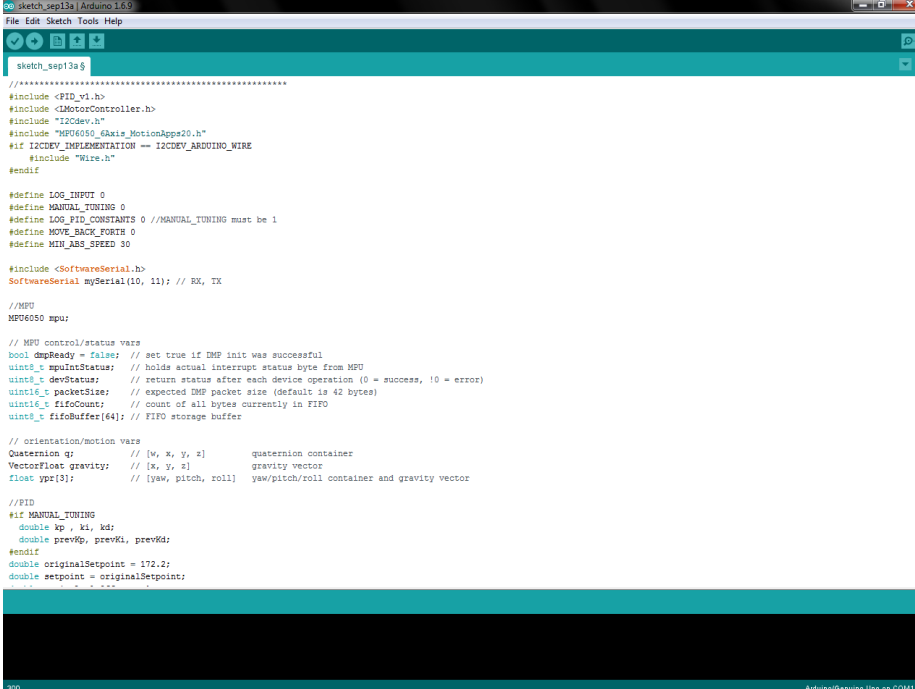
void readPIDTuningValues()
{
  int potKp = analogRead(A0);
  int potKi = analogRead(A1);
  int potKd = analogRead(A2);

  kp = map(potKp, 0, 1023, 0, 25000) / 100.0; //0 - 250
  ki = map(potKi, 0, 1023, 0, 100000) / 100.0; //0 - 1000
  kd = map(potKd, 0, 1023, 0, 500) / 100.0; //0 - 5
}
#endif

//*****

```

- d. Proses berikutnya adalah melakukan *Verify/Compile* program dan *Upload* program, dengan memilih menu *Sketch -> Upload* pada *software* Arduino 1.6.9, seperti pada gambar di bawah berikut ini :



```

sketch_sep13a | Arduino IDE
File Edit Sketch Tools Help

sketch_sep13a$
//*****
#include <I2Cdev.h>
#include <MotorController.h>
#include "I2Cdev.h"
#include "MPU6050_Axis_MotionApps20.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif

#define LOG_INPUT 0
#define MANUAL_TUNING 0
#define LOG_FID_CONSTANTS 0 //MANUAL_TUNING must be 1
#define MOVE_BACK_FORITS 0
#define MPH_ABS_SPEED 30

#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11); // RX, TX

//MPU
MPU6050 mpu;

// MPU control/status vars
bool dmpReady = false; // set true if DMP init was successful
uint8_t mpuIntStatus; // holds actual interrupt status byte from MPU
uint8_t devStatus; // return status after each device operation (0 = success, != 0 = error)
uint16_t packetSize; // expected DMP packet size (default is 42 bytes)
uint16_t fifoCount; // count of all bytes currently in FIFO
uint8_t fifoBuffer[64]; // FIFO storage buffer

// orientation/motion vars
Quaternion q; // [w, x, y, z] quaternion container
VectorFloat gravity; // [x, y, z] gravity vector
float ypr[3]; // [yaw, pitch, roll] yaw/pitch/roll container and gravity vector

//PID
#if MANUAL_TUNING
double kp, ki, kd;
double prevKp, prevKi, prevKd;
#endif
double originalSetpoint = 172.2;
double setpoint = originalSetpoint;

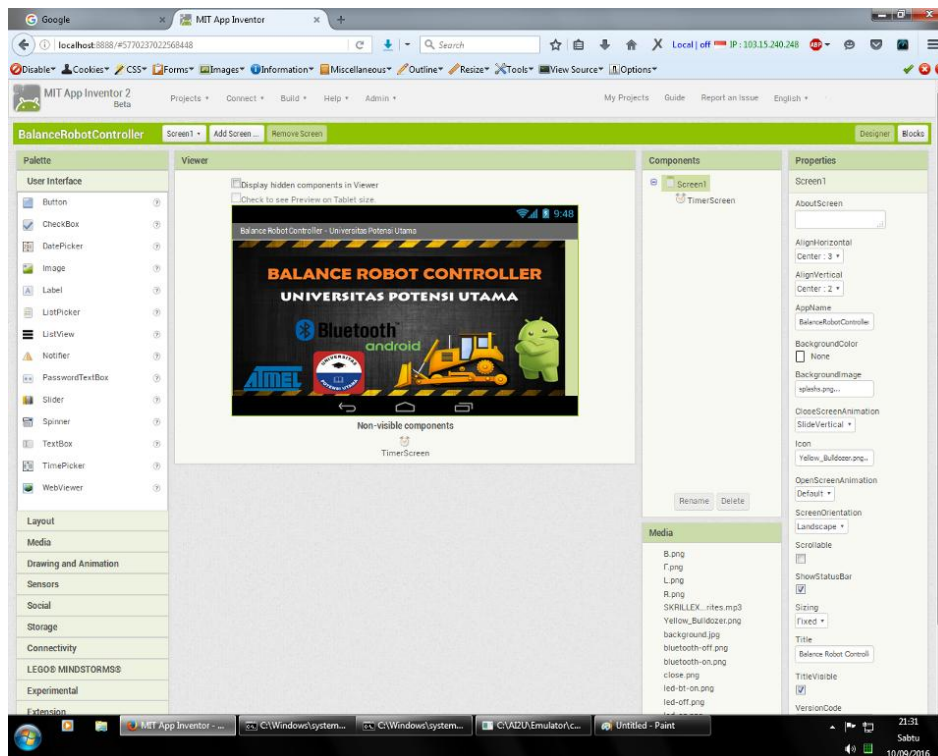
```

**Gambar IV.5. Proses Upload Program Software Arduino 1.6.9**

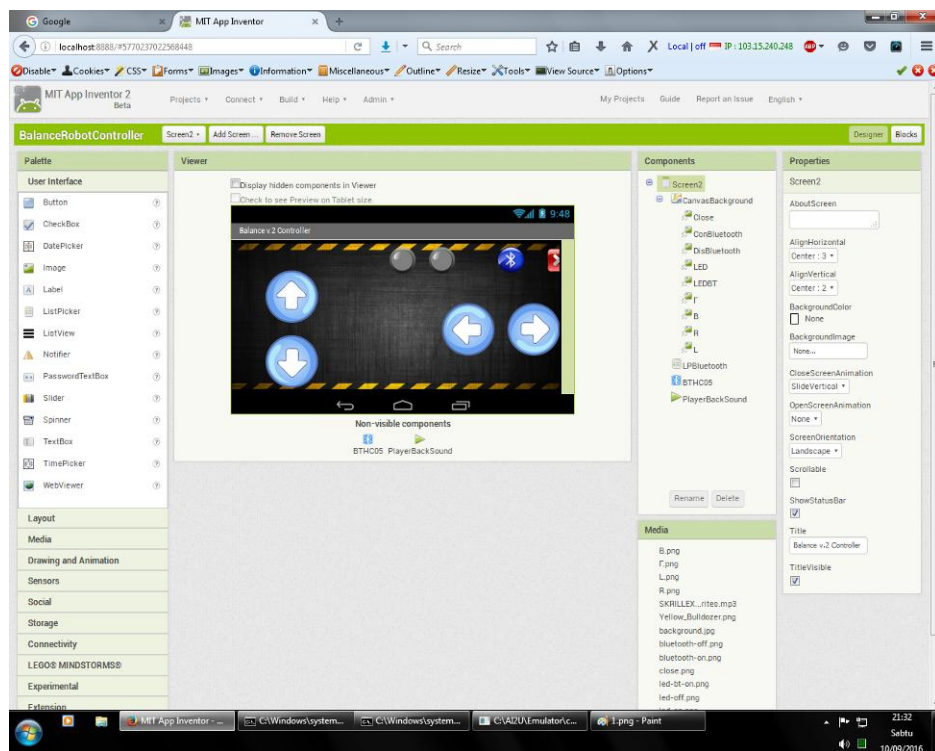
- e. Setelah proses *upload* program selesai terhadap rangkaian mikrokontroler, maka dapat dilihat kinerja dari perangkat berjalan sesuai dengan program yang diperintahkan dengan melakukan pengujian perangkat secara *hardware*.

Pengujian *software* berikutnya adalah perancangan aplikasi pengendali robot penyeimbang menggunakan sensor jarak berbasis android. Perancangan menggunakan *software app inventor 2 ultimate*. Berikut adalah hasil dari perancangan dari aplikasi ditampilkan pada gambar berikut :

## A. Tampilan Awal

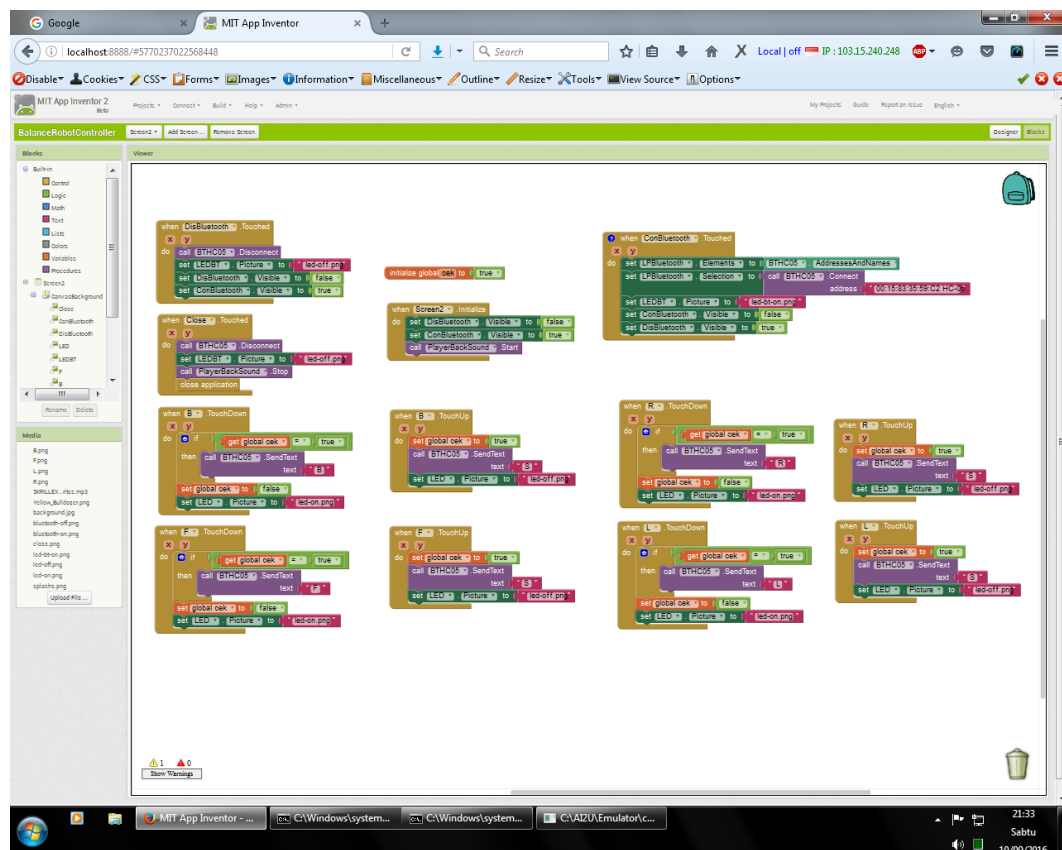


## B. Tampilan Utama



Gambar IV.6. Perancangan *Software* Aplikasi Pengendali

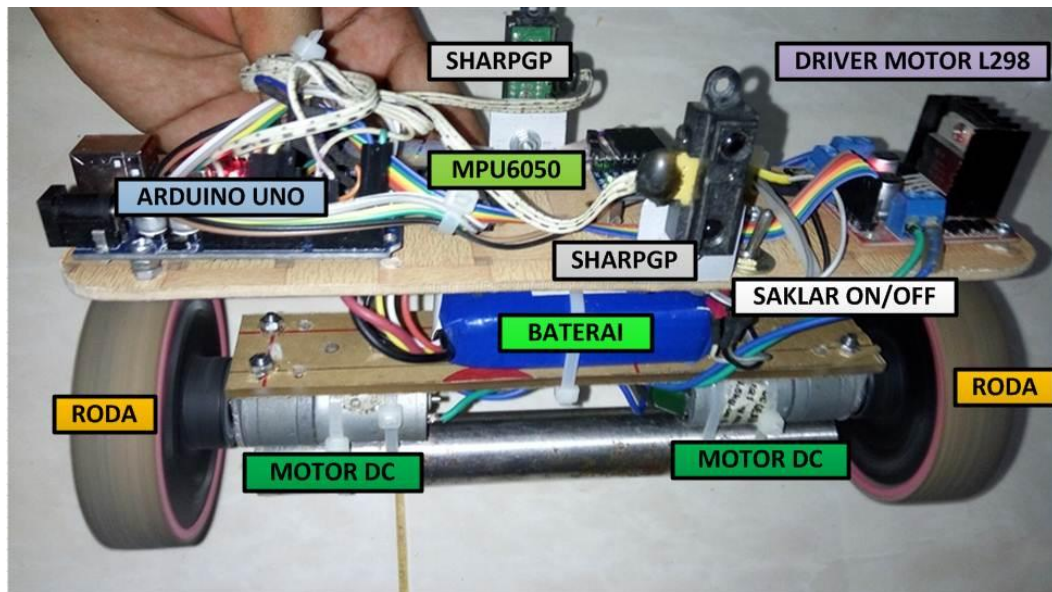
Untuk *listing* program aplikasi pengendali, berikut adalah gambar dari blok dari aplikasi aplikasi pengendali pada *smartphone* android, ditunjukkan pada gambar di bawah ini :



**Gambar IV.7. Blok Program *Software* Aplikasi Pengendali**

#### IV.5. Pengujian *Hardware*

Setelah semua rangkaian yang telah selesai dirancang pada perangkat robot penyeimbang menggunakan sensor jarak berbasis android, kemudian dilakukan penyatuan semua rangkaian yang telah selesai dengan mekanik. Berikut adalah gambar hasil dari perancangan sistem mekanik dan eletronik ditunjukkan oleh gambar IV.8 Berikut :



**Gambar IV.8. Hardware Mekanik dan Elektronik**

#### IV.6. Uji Coba Perangkat

Setelah semua komponen terpasang dan program selesai disusun, maka langkah berikutnya adalah melakukan pengujian alat. Pengujian ini dilakukan secara bertahap dari rangkaian ke rangkaian berikutnya.

##### IV.6.1. Pengujian Rangkaian Mikrokontroler Arduino Uno

Untuk mengetahui apakah rangkaian mikrokontroler Arduino Uno telah bekerja dengan baik, maka dilakukan pengujian. Pengujian bagian ini dilakukan dengan memberikan program pada mikrokontroler Arduino Uno. Program sederhana yang digunakan adalah *blink led* pada Pin 13 Arduino. Berikut adalah listing program dari *blink led* :

```

void setup() {
// initialize digital pin 13 as an output.
pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
digitalWrite(13, HIGH); // turn the LED on
delay(1000);           // wait for a second
digitalWrite(13, LOW); // turn the LED off
delay(1000);           // wait for a second
}

```

Jika setelah *upload* program dilakukan dan led berkedip setiap 1000 milisekon (1 detik) maka arduino dalam keadaan baik.

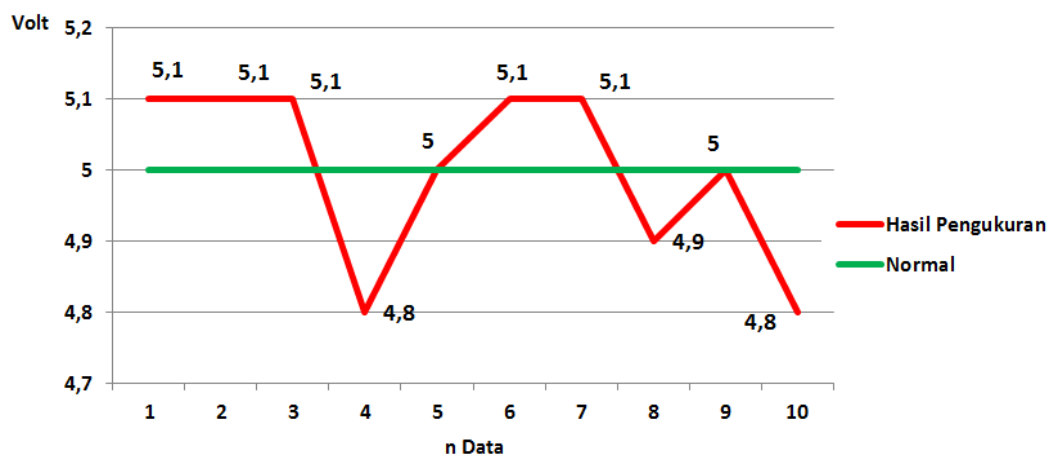
#### IV.6.3. Pengujian Rangkaian Regulator Tegangan

Pengujian ini dilakukan untuk mengukur tegangan yang dihasilkan dari regulator tegangan LM7805. Regulator mengubah tegangan dari baterai menjadi 5VDC untuk tegangan kerja perangkat keseluruhan. Pengukuran dilakukan menggunakan multimeter atau *voltmeter*. Berikut adalah hasil dari pengukuran tegangan, ditunjukkan pada tabel IV.1 :

**Tabel IV.1. Hasil Pengujian Regulator Tegangan 5VDC**

No. Pengujian	Hasil Pengukuran (Volt)	Error (Volt)
1	5,1	0,1
2	5,1	0,1
3	5,1	0,1
4	4,8	0,2
5	5	0
6	5,1	0,1
7	5,1	0,1
8	4,9	0,1
9	5	0
10	4,8	0,2
<b>Σ Error</b>		<b>1,0</b>
<b>Rata - Rata Error</b>		<b>0,1</b>

Berdasarkan data dari tabel di atas, disimpulkan bahwa *error* dari tegangan normal dengan tegangan regulator 5VDC memiliki total selisih *error*  $\pm 1$  Volt pada 10 kali pengujian (n) atau rata – rata *error* sebesar 0,1 Volt. Berdasarkan tabel diatas, dapat digambarkan pada grafik di bawah ini :



**Gambar IV.9. Grafik Hasil Perbandingan Tegangan Normal dengan Regulator 5VDC**

Terlihat hasil kedua pengujian hampir sama dengan selisih nilai pengukuran yang kecil. Kesimpulan dari pengujian ini adalah regulator tegangan 5VDC dapat dinyatakan berkerja dengan baik dan sesuai jika dibandingkan dengan tegangan kerja normal yaitu 5VDC.

#### IV.6.3. Pengujian Rangkaian Sensor Jarak Sharpgp GP11

SRF04 sebagai sensor jarak yang mengirimkan data jarak terhadap objek. Pengujian ini dilakukan dengan mencoba mengirimkan data jarak ke mikrokontroler. Data pembanding yang digunakan adalah penggaris dengan pengambilan data 10 cm – 50 cm. Berikut *listing* program untuk menampilkan data sensor jarak Sharpgp GP11 dalam cm :

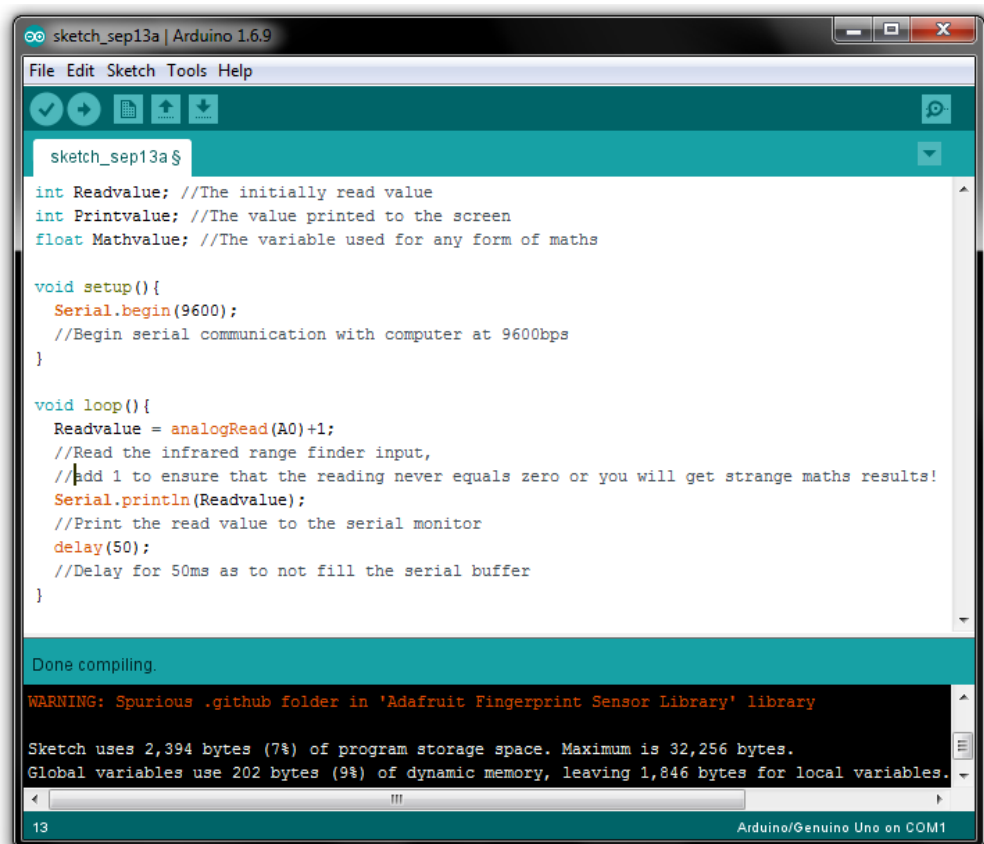
```

int Readvalue; //The initially read value
int Printvalue; //The value printed to the screen
float Mathvalue; //The variable used for any form of maths

void setup(){
  Serial.begin(9600);
  //Begin serial communication with computer at 9600bps
}

void loop(){
  Readvalue = analogRead(A0)+1;
  //Read the infrared range finder input,
  //add 1 to ensure that the reading never equals zero or you will get strange
  maths results!
  Serial.println(Readvalue);
  //Print the read value to the serial monitor
  delay(50);
  //Delay for 50ms as to not fill the serial buffer
}

```



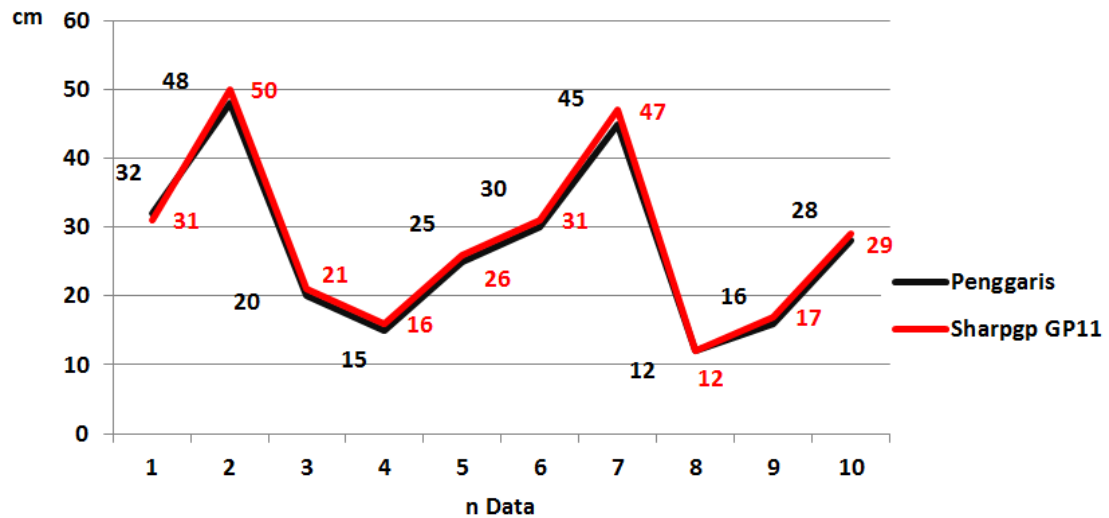
**Gambar IV.10. Listing Program Pengujian Sharpgp GP11**

Berikut adalah hasil dari pengukuran sensor jarak terhadap penggaris, ditunjukkan pada tabel IV.2 :

**Tabel IV.2. Hasil Pengujian Sensor Sharpgp GP11 Terhadap Penggaris**

No. Pengujian	Hasil Pengukuran (cm)		Error (cm)
	Penggaris	Sharpgp GP11	
1	32	31	1
2	48	50	2
3	20	21	1
4	15	16	1
5	25	26	1
6	30	31	1
7	45	47	2
8	12	12	1
9	16	17	1
10	28	29	1
<b><math>\Sigma</math> Error</b>			<b>12</b>
<b>Rata - Rata Error</b>			<b>1,2</b>

Berdasarkan data dari tabel di atas, disimpulkan bahwa *error* dari perhitungan jarak penggaris dengan Sharpgp GP11 memiliki total selisih *error*  $\pm$  12 cm pada 10 kali pengujian (n) atau rata – rata *error* sebesar 1,2 cm. Berdasarkan tabel diatas, dapat digambarkan pada grafik di bawah ini :



**Gambar IV.11. Grafik Hasil Perbandingan Sensor Jarak Sharp GP11 Terhadap Penggaris**

#### IV.6.4. Pengujian Transmisi Data Bluetooth HC-05

Pengukuran jarak transmisi bertujuan untuk mengetahui seberapa jauh *bluetooth* dapat berhubungan dan mampu membawa perintah dari *smartphone* ke mikrokontroler. jarak jangkauan maksimum *bluetooth* adalah 10 meter. Pengujian dilakukan dengan dua metode yaitu pengujian jarak tanpa halangan dan pengujian jarak dengan banyak halangan untuk menghambat transmisi data, seperti lemari dan perabotan rumah lainnya. Tabel IV.3. berikut adalah tabel dari hasil pengujian tanpa halangan.

**Tabel IV.3. Hasil Pengujian Jarak Transmisi Tanpa Halangan**

Jarak	Pengujian I	Pengujian II	Pengujian III	Pengujian IV	Pengujian V
1 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
2 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
3 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
4 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
5 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
6 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim

7 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
8 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
9 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
10 Meter	Tidak Terkirim	Terkirim	Tidak Terkirim	Terkirim	Terkirim
<b>Error</b>	<b>1%</b>				

Pengujian jarak tanpa halangan dilakukan sebanyak 5 kali tahap pengujian dengan jarak maksimum sejauh 10 meter. Dari hasil pengujian, pada saat perangkat mencapai jarak jangkauan maksimum 10 meter, perintah yang dikirimkan *smartphone* tidak diterima perangkat, dari 50 kali pengiriman perintah terdapat 2 kali gagal diterima atau sebanyak 1%.

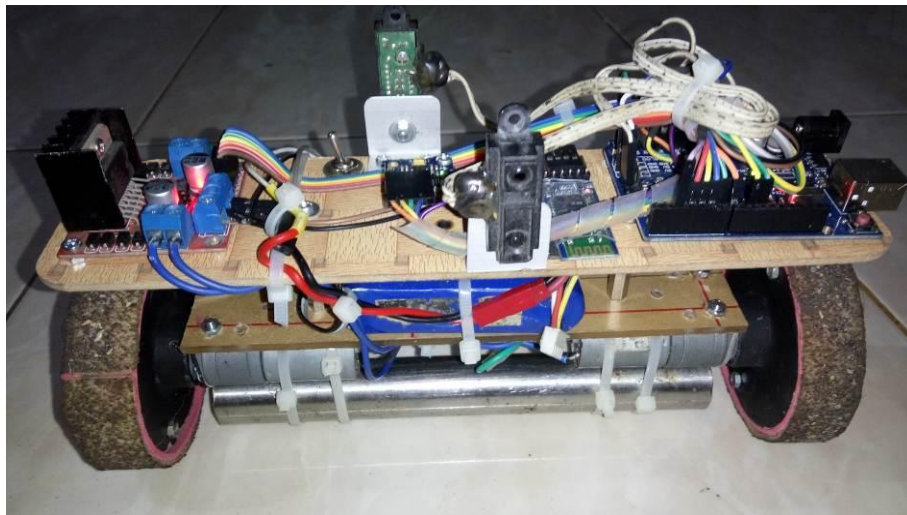
**Tabel IV.4. Hasil Pengujian Jarak Transmisi Dengan Halangan**

Jarak	Pengujian I	Pengujian II	Pengujian III	Pengujian IV	Pengujian V
1 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
2 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
3 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
4 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
5 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
6 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
7 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
8 Meter	Terkirim	Terkirim	Terkirim	Terkirim	Terkirim
9 Meter	Tidak Terkirim	Terkirim	Tidak Terkirim	Tidak Terkirim	Terkirim
10 Meter	Terkirim	Terkirim	Tidak Terkirim	Tidak Terkirim	Terkirim
<b>Error</b>	<b>2.5%</b>				

Sedangkan pengujian jarak dengan halangan dilakukan sebanyak 5 kali tahap pengujian dengan jarak maksimum sejauh 10 meter, terdapat perintah yang dikirimkan *smartphone* tidak diterima perangkat sebanyak 5 kali gagal diterima atau sebanyak 2.5%.

#### IV.6.5. Pengujian Analisa Perangkat Keseluruhan

Pengujian ini dilakukan untuk mengetahui apakah perancangan perancangan robot penyeimbang menggunakan sensor jarak berbasis android bekerja sesuai dengan logika program. Sebelum pengujian dilakukan, perangkat dalam telah menyala dan keadaan robot dalam keadaan normal. Keadaan normal yang dimaksud adalah keadaan dimana robot menyeimbangkan diri. Berikut adalah gambar dari kondisi robot ketika dalam kondisi normal :




**Gambar IV.12. Kondisi Normal Robot Penyeimbang**

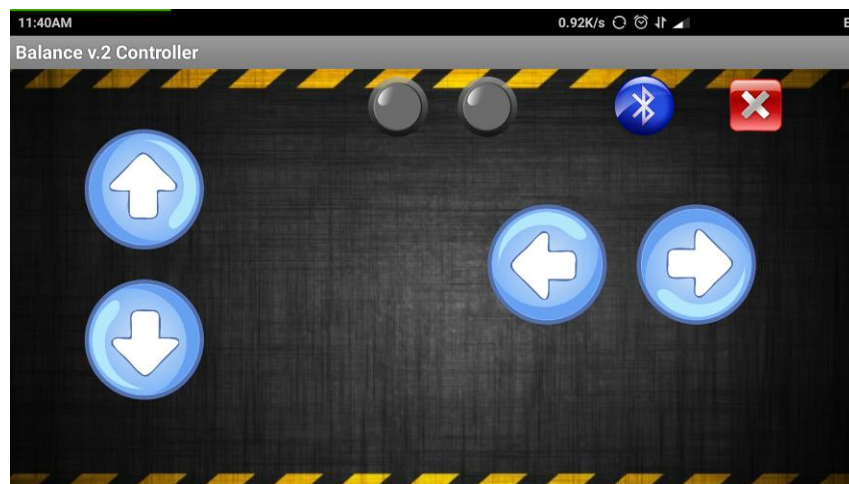
Pengujian berikutnya adalah mencoba koneksi *bluetooth* perangkat robot dengan perangkat *smartphone* android. Proses awal adalah menjalankan aplikasi seperti gambar berikut :



**Gambar IV.13. Kondisi Smartphone Menjalankan Aplikasi**

Koneksi dapat dilakukan dengan menekan tombol “” pada aplikasi.

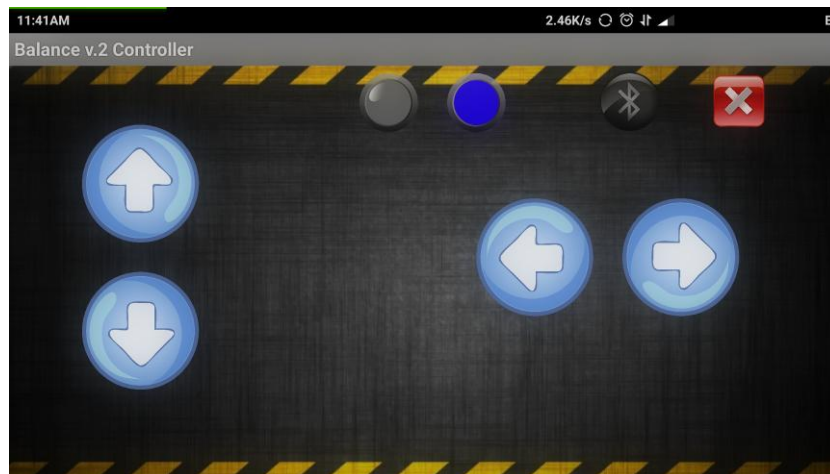
Perangkat akan otomatis terkoneksi. Berikut adalah gambar proses melakukan koneksi terhadap perangkat robot.



**Gambar IV.14. Kondisi Smartphone Melakukan Koneksi.**

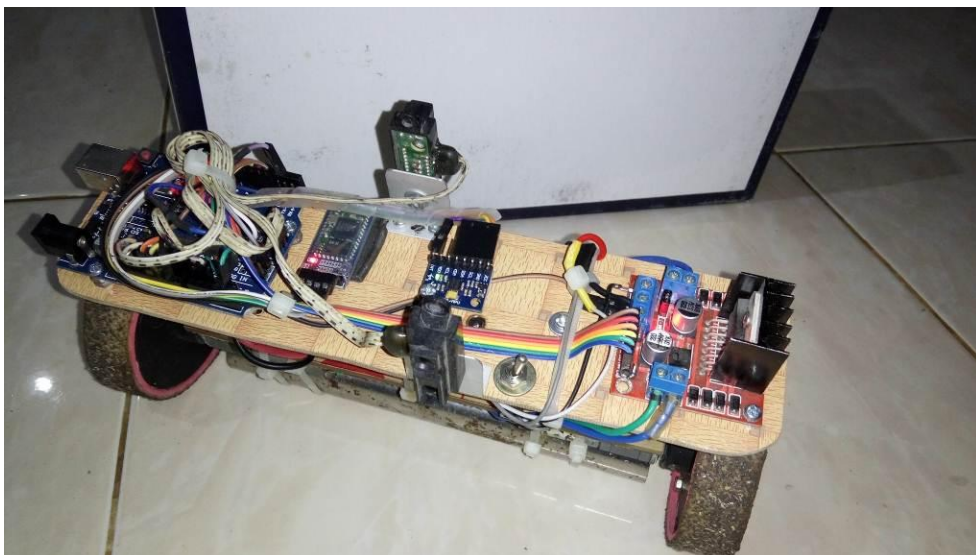
Jika koneksi berhasil, perangkat akan menampilkan indikator menyala.

Berikut adalah gambar dari koneksi berhasil.

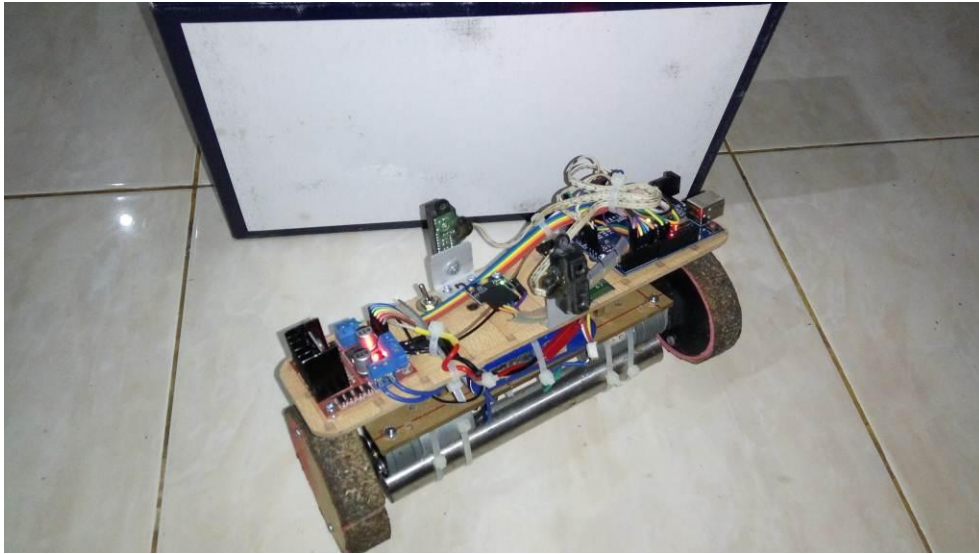


**Gambar IV.15. Kondisi Smartphone Berhasil Melakukan Koneksi.**


Robot penyeimbang akan mengikuti pergerakan sesuai dengan data yang dikirimkan oleh aplikasi android. Robot juga merespon objek yang terdeteksi pada sensor jarak posisi depan ataupun sensor jarak belakang. Berikut adalah gambar hasil pengujian bahwa robot bergerak mengikuti perintah dari pengguna dan mendeteksi objek sekitar.




**Gambar IV.16. Robot Bergerak Menyesuaikan Perintah (1).**



**Gambar IV.17. Robot Bergerak Menyesuaikan Perintah (2).**

Untuk melakukan diskoneksi, tekan kembali tombol ". Aplikasi

akan berhenti jika pengguna menekan tombol ".

#### **IV.7. Kelebihan dan Kekurangan**

Pada perancangan robot penyeimbang menggunakan sensor jarak berbasis android ini masih kurang sempurna. Perakitan dan pembuatan perangkat ini masih memiliki beberapa kelebihan dan kekurangan, diantaranya:

##### **a. Kelebihan**

Adapun beberapa kelebihan yang dimiliki perancangan robot penyeimbang menggunakan sensor jarak berbasis android ini, antara lain :

1. Dengan adanya perangkat ini, maka kita dapat mengendalikan robot bergerak maju, mundur, kanan dan kiri sesuai dengan keinginan pengguna dalam keadaan seimbang dan menghindari objek yang terdeteksi oleh sensor jarak.

2. Robot tetap dapat menyeimbangkan diri walaupun diberikan dorongan dari luar.
3. Perangkat robot *balancing* bekerja menggunakan baterai lipo 11.1V 1A dan regulator tegangan 5VDC.

**b. Kekurangan**

Adapun beberapa kekurangan yang dimiliki perancangan robot penyeimbang menggunakan sensor jarak berbasis android antara lain:

1. Perancangan mekanik robot cukup berat, sehingga konsumsi tegangan menjadi besar.
2. Respon robot terhadap data yang dikirimkan aplikasi terdapat *delay* (tunda) yang menyebabkan sering terjadi kehilangan data dan robot bergerak dalam kondisi *error*.
3. Sensor jarak hanya dapat mendeteksi objek pada posisi depan dan belakang dari perangkat (robot).
4. Aplikasi yang dirancang telah disesuaikan dengan *module bluetooth* HC-05, sehingga aplikasi tidak dapat digunakan untuk mengontrol robot lainnya.
5. Sumber daya yang dibutuhkan untuk mengendalikan motor DC cukup besar dan terjadi pemanasan berlebih pada regulator tegangan.



## **BAB V**

### **KESIMPULAN DAN SARAN**

## BAB V

### KESIMPULAN DAN SARAN

#### V.1. Kesimpulan

Setelah perancangan robot penyeimbang menggunakan sensor jarak berbasis android ini dilakukan pengujian dan analisa, sehingga didapatkan kesimpulan sebagai berikut :

4. Perancangan mekanik robot menggunakan 2 roda dan bergerak menyeimbangkan diri. Penggunaan Arduino Uno sebagai pengendali utama, motor DC sebagai *output* penggerak mekanik, Sensor jarak Sharpgp GP11 untuk mendeteksi objek.
5. Dengan adanya perangkat ini, maka kita dapat menggunakan Android untuk mengendalikan robot bergerak maju, mundur, kanan dan kiri sesuai dengan keinginan pengguna dalam keadaan seimbang, robot tetap dapat menyeimbangkan diri walaupun diberikan dorongan dari luar dan menghindari objek yang terdeteksi oleh sensor jarak.
6. Dari hasil pengujian, sensor jarak Sharpgp GP11 yang digunakan akurat dibandingkan dengan data pengukuran menggunakan penggaris dan *module bluetooth* dapat bekerja dengan baik.
7. Perangkat robot *balancing* bekerja menggunakan baterai lipo 11.1V 1A dan regulator tegangan 5VDC. Perancangan mekanik robot cukup berat, sehingga konsumsi tegangan menjadi besar dan penggunaan baterai tidak bertahan lama.

8. Tingkat keakuratan pengiriman sangat tinggi pada jarak jangkauan dibawah 8 meter dan gangguan seperti gagalnya pengiriman perintah terjadi pada jarak 9 – 10 meter.
9. Respon robot terhadap data yang dikirimkan aplikasi terdapat *delay* (tunda) yang menyebabkan sering terjadi kehilangan data dan robot bergerak dalam kondisi *error*.
10. Hanya bisa dikendalikan pada satu pengendali saja dalam waktu yang bersamaan. Aplikasi yang dirancang telah disesuaikan dengan *module bluetooth* HC-05, sehingga aplikasi tidak dapat digunakan untuk mengontrol robot lainnya.

## V.2. Saran

Dalam perancangan robot penyeimbang menggunakan sensor jarak berbasis android terdapat beberapa kendala yang dihadapi penulis. Maka penulis akan menyampaikan beberapa saran yang diharapkan pembaca dapat memahami prinsip perangkat yang dirancang sehingga dapat mengembangkan skripsi ini. Adapun saran – saran tersebut adalah:

1. Pengendalian ini masih menggunakan aplikasi yang dapat menampilkan komunikasi 1 arah, untuk pengembangan selanjutnya diharapkan dapat menyempurnakan aplikasi sehingga dapat berkomunikasi 2 arah.
2. Dari segi komunikasi, perangkat hanya dapat mengirimkan perintah menggunakan *bluetooth* sehingga jarak jangkauan maksimum 10 meter, untuk pengembangan lebih lanjut dapat menggunakan komunikasi *wifi* sehingga dapat digunakan secara jarak jauh.

3. Untuk pengembangan lebih lanjut, pengendali dapat menggunakan semua perangkat *smartphone*, *tablet* PC atau perangkat lainnya yang berbasis sistem operasi *android*.



**LAMPIRAN**

```

//*****
#include <PID_v1.h>
#include <LMotorController.h>
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    #include "Wire.h"
#endif

#define LOG_INPUT 0
#define MANUAL_TUNING 0
#define LOG_PID_CONSTANTS 0 //MANUAL_TUNING must be 1
#define MOVE_BACK_FORTH 0
#define MIN_ABS_SPEED 30

#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11); // RX, TX

//MPU
MPU6050 mpu;

// MPU control/status vars
bool dmpReady = false; // set true if DMP init was successful
uint8_t mpuIntStatus; // holds actual interrupt status byte from MPU
uint8_t devStatus; // return status after each device operation (0 = success, != 0 = error)
uint16_t packetSize; // expected DMP packet size (default is 42 bytes)
uint16_t fifoCount; // count of all bytes currently in FIFO
uint8_t fifoBuffer[64]; // FIFO storage buffer

// orientation/motion vars
Quaternion q; // [w, x, y, z] quaternion container
VectorFloat gravity; // [x, y, z] gravity vector
float ypr[3]; // [yaw, pitch, roll] yaw/pitch/roll container and gravity vector

//PID
#if MANUAL_TUNING
    double kp , ki, kd;
    double prevKp, prevKi, prevKd;
#endif
double originalSetpoint = 172.2;
double setpoint = originalSetpoint;
double movingAngleOffset = 1;
double input, output;
int moveState=0; //0 = balance; 1 = back; 2 = forth
char perintah;

```

```

#if MANUAL_TUNING
  PID pid(&input, &output, &setpoint, 0, 0, 0, DIRECT);
#else
  PID pid(&input, &output, &setpoint, 20, 80, 1.3, DIRECT);
#endif

//MOTOR CONTROLLER
int ENA = 3;
int IN1 = 4;
int IN2 = 8;
int IN3 = 5;
int IN4 = 7;
int ENB = 6;
LMotorController motorController(ENA, IN1, IN2, ENB, IN3, IN4, 0.6, 1);

//timers
long time1Hz = 0;
long time5Hz = 0;

volatile bool mpuInterrupt = false; // indicates whether MPU interrupt pin has
gone high
void dmpDataReady()
{
  mpuInterrupt = true;
}

void setup()
{
  //motorController.move(150,150,10);
  //delay(1000);
  //motorController.move(0,0,10);
  ///delay(1000);
  //motorController.move(-150,-150,10);
  ///delay(1000);
  //motorController.move(0,0,10);
  //delay(1000);
  // join I2C bus (I2Cdev library doesn't do this automatically)
  #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    Wire.begin();
    TWBR = 24; // 400kHz I2C clock (200kHz if CPU is 8MHz)
  #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
    Fastwire::setup(400, true);
  #endif
}

```

```

// initialize serial communication
// (115200 chosen because it is required for Teapot Demo output, but it's
// really up to you depending on your project)
Serial.begin(115200);
mySerial.begin(9600);
while (!Serial); // wait for Leonardo enumeration, others continue immediately

// initialize device
Serial.println(F("Initializing I2C devices..."));
mpu.initialize();

// verify connection
Serial.println(F("Testing device connections..."));
Serial.println(mpu.testConnection() ? F("MPU6050 connection successful") :
F("MPU6050 connection failed"));

// load and configure the DMP
Serial.println(F("Initializing DMP..."));
devStatus = mpu.dmpInitialize();

// supply your own gyro offsets here, scaled for min sensitivity
mpu.setXGyroOffset(682);
mpu.setYGyroOffset(524);
mpu.setZGyroOffset(-271);
mpu.setZAccelOffset(25816); // 1688 factory default for my test chip

// make sure it worked (returns 0 if so)
if (devStatus == 0)
{
  // turn on the DMP, now that it's ready
  Serial.println(F("Enabling DMP..."));
  mpu.setDMPEnabled(true);

  // enable Arduino interrupt detection
  Serial.println(F("Enabling interrupt detection (Arduino external interrupt
0)..."));
  attachInterrupt(0, dmpDataReady, RISING);
  mpuIntStatus = mpu.getIntStatus();

  // set our DMP Ready flag so the main loop() function knows it's okay to use
it
  Serial.println(F("DMP ready! Waiting for first interrupt..."));
  dmpReady = true;

  // get expected DMP packet size for later comparison
  packetSize = mpu.dmpGetFIFOPacketSize();

```

```

    //setup PID
    pid.SetMode(AUTOMATIC);
    pid.SetSampleTime(10);
    pid.SetOutputLimits(-255, 255);
}
else
{
    // ERROR!
    // 1 = initial memory load failed
    // 2 = DMP configuration updates failed
    // (if it's going to break, usually the code will be 1)
    Serial.print(F("DMP Initialization failed (code ");
    Serial.print(devStatus);
    Serial.println(F(""));
}
}

void loop()
{
    // if programming failed, don't try to do anything
    if (!dmpReady) return;

    // wait for MPU interrupt or extra packet(s) available
    while (!mpuInterrupt && fifoCount < packetSize)
    {
        //no mpu data - performing PID calculations and output to motors
        pid.Compute();
        motorController.move(output, MIN_ABS_SPEED);

        unsigned long currentMillis = millis();
        if (currentMillis - time1Hz >= 1000)
        {
            loopAt1Hz();
            time1Hz = currentMillis;
        }

        if (currentMillis - time5Hz >= 5000)
        {
            loopAt5Hz();
            time5Hz = currentMillis;
        }
    }

    // reset interrupt flag and get INT_STATUS byte
    mpuInterrupt = false;

```

```

mpuIntStatus = mpu.getIntStatus();

// get current FIFO count
fifoCount = mpu.getFIFOCount();

// check for overflow (this should never happen unless our code is too
inefficient)
if ((mpuIntStatus & 0x10) || fifoCount == 1024)
{
    // reset so we can continue cleanly
    mpu.resetFIFO();
    Serial.println(F("FIFO overflow!"));

// otherwise, check for DMP data ready interrupt (this should happen
frequently)
}
else if (mpuIntStatus & 0x02)
{
    // wait for correct available data length, should be a VERY short wait
    while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount();

    // read a packet from FIFO
    mpu.getFIFOBytes(fifoBuffer, packetSize);

    // track FIFO count here in case there is > 1 packet available
    // (this lets us immediately read more without waiting for an interrupt)
    fifoCount -= packetSize;

    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
    #if LOG_INPUT
        Serial.print("ypr\t");
        Serial.print(ypr[0] * 180/M_PD);
        Serial.print("\t");
        Serial.print(ypr[1] * 180/M_PI);
        Serial.print("\t");
        Serial.println(ypr[2] * 180/M_PI);
    #endif
    input = ypr[1] * 180/M_PI + 180;

    if (mySerial.available()) {
perintah = mySerial.read();
if(perintah=='F'){
Serial.println("F");
motorController.move(150,150,10); delay(50);

```



```

}


//PID Tuning (3 potentiometers)
#if MANUAL_TUNING
void setPIDTuningValues()
{
  readPIDTuningValues();
  if (kp != prevKp || ki != prevKi || kd != prevKd)
  {
    #if LOG_PID_CONSTANTS
      Serial.print(kp);Serial.print("
      ");Serial.print(ki);Serial.print("
      ");Serial.println(kd);
    #endif
    pid.SetTunings(kp, ki, kd);
    prevKp = kp; prevKi = ki; prevKd = kd;
  }
}

void readPIDTuningValues()
{
  int potKp = analogRead(A0);
  int potKi = analogRead(A1);
  int potKd = analogRead(A2);

  kp = map(potKp, 0, 1023, 0, 25000) / 100.0; //0 - 250
  ki = map(potKi, 0, 1023, 0, 100000) / 100.0; //0 - 1000
  kd = map(potKd, 0, 1023, 0, 500) / 100.0; //0 - 5
}
#endif

//*****

```

	<b>DOKUMEN LEVEL FORM</b>	<b>NO. DOKUMEN</b> F-FTIK-12-08
<b>JUDUL</b> SURAT PENGAJUAN JUDUL SKRIPSI		Tanggal Terbit : 07 Nov 2014 Tanggal Efektif : 14 Nov 2014
<b>AREA PROGRAM STUDI</b>		Halaman : 44 dari 48 <b>NO.REVISI</b> 00

Medan, 23 Desember 2015  
Hal : Pengajuan Judul Skripsi  
Lamp : -  
Kepada Yth,  
Ketua Program Studi Teknik Informatika  
di  
Medan

Dengan hormat,

Yang bertanda tangan dibawah ini :

NIM : 1110000271  
Nama : Muhammad Nazri  
Program Studi : Teknik Informatika  
Peminatan : Kecerdasan Buatan

Mengajukan Judul Skripsi sebagai berikut :

1. Implementasi Pembayaran Uang Kuliah Berbasis RFID Dengan Interface
2. Perancangan Robot Penyeimbang Menggunakan Sensor Jarak Berbasis Android.

Atas perhatiaanya saya ucapakan terima kasih.

Pemohon



Muhammad Nazri  
Nim. 1110000271

Judul Skripsi yang disetujui No 2 / tanggal 23-12-2015  
Nama Pembimbing : I. Edy Victor Haryanto, M.Kom  
II. Iwan Fitrianto Rahmad, M.Kom

Ketua Program Studi

(Budi Triandi, M.Kom)

- Dibuat tanggal 23/12/2015
1. Program Studi- TI
  2. Mahasiswa
  3. Pembimbing I
  4. Pembimbing II



# UNIVERSITAS POTENSI UTAMA FAKULTAS TEKNIK DAN ILMU KOMPUTER

SK. Mendikbud R.I. No.: 424/E/O/2014

Kampus : Jl. K.L. Yos Sudarso KM 6,5 No. 3-A Telp : (061) 6640525 Fax : (061) 6636830 Tanjung Mulia-Medan  
Website : <http://www.potensi-utama.ac.id>  
E-mail : [info@potensi-utama.ac.id](mailto:info@potensi-utama.ac.id)

## FORMULIR PENDAFTARAN JUDUL SKRIPSI

### I. UMUM [Diisi oleh mahasiswa]

Nama Mahasiswa : MUHAMMAD NAZZI  
NIM : 111.0000.271  
Program Studi : TEKNIK INFORMATIKA  
Nama Dosen Wali : BAMBANG T. HUTAGALUNG

### II. PERSYARATAN PENGAMBILAN SKRIPSI : [Diperiksa oleh Ka Prodi/Sek Prodi]

- Sudah Lulus Praktek Kerja Lapangan:  
 Ya  Tidak
- Sudah Menjalani Kuliah Minimum 137 SKS dari Total 146 SKS untuk Kurikulum 2008  
 Ya  Tidak
- Mengambil Kredit Mata Kuliah Skripsi:  
 Ya  Tidak
- Sudah Membuat Proposal Judul Skripsi:  
 Ya  Tidak

[Ketentuan: Persyaratan harus dipenuhi]


### III. DATA SKRIPSI :

- Judul : [Diisi oleh mahasiswa]  
PERANCANGAN ROBOT PENJEMBANG MENGGUNAKAN SENSOR  
JARAK BERBASIS ANDROID

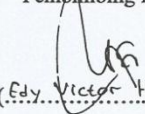
[Diisi oleh Bagian Program Studi]

- Pembimbing I : EDY VICTOR HARYANTO, M.KOM
- Pembimbing II : IWAN FITRIANTO RAHMAN, M.KOM

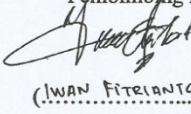
Medan, .....

Mengetahui Ketua Prodi  
  
(Badi Triandi, M.Kom)


Pembimbing I,


  
(Edy Victor H. M. Kom)

Pembimbing II

  
(IWAN FITRIANTO R. M. Kom)

Diterima oleh Bagian Program Studi Tanggal : 31.08.2016

  
(.....)

	<b>DOKUMEN LEVEL FORM</b>	<b>NO. DOKUMEN F-FTIK-12-10</b>
<b>JUDUL</b> SURAT PERNYATAAN KESEDIAAN PEMBIMBING SKRIPSI		Tanggal Terbit : 07 Nov 2014
<b>AREA PROGRAM STUDI</b>		Tanggal Efektif : 14 Nov 2014
		Halaman : 1 dari 1
		<b>NO. REVISI 00</b>

Saya yang bertanda tangan dibawah ini

Nama : Edy Victor Haryanto, M.Kom

Pangkat/ Golongan : -

Jabatan : Kabag LPPM

Alamat : Martubung Jl. Kepiting No. 166

Dengan ini menyatakan kesedian saya untuk memberikan bimbingan skripsi atas nama mahasiswa berikut:

Nama : Muhammad Nazri


NIM : 1110000271


Program Studi : Teknik Informatika

Jenjang Pendidikan : Strata-1 (S-1)

Demikian surat pernyataan ini diperbuat dengan sebenarnya untuk dapat digunakan seperlunya

Medan, 9 April 2015

  
(Edy Victor Haryanto, M.Kom)

	<b>DOKUMEN LEVEL FORM</b>	<b>NO. DOKUMEN F-FTIK-12-10</b>
<b>JUDUL</b> SURAT PERNYATAAN KESEDIAAN PEMBIMBING SKRIPSI		Tanggal Terbit : 07 Nov 2014 Tanggal Efektif : 14 Nov 2014
<b>AREA PROGRAM STUDI</b>		Halaman : 1 dari 2 <b>NO. REVISI 00</b>

Saya yang bertanda tangan dibawah ini

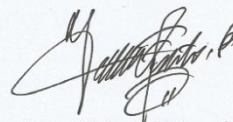
Nama : Iwan Fitrianto Rahmad, M.Kom  
Pangkat/ Golongan : -  
Jabatan : Lektor  
Alamat : Jl. Rawe III Lorong Tengah Gg. Restu No.137

Dengan ini menyatakan kesedian saya untuk memberikan bimbingan skripsi atas nama mahasiswa berikut:

Nama : Muhammad Nazri  
NIM : 1110000271  
Program Studi : Teknik Informatika  
Jenjang Pendidikan : Strata-1 (S-1)

Demikian surat pernyataan ini diperbuat dengan sebenarnya untuk dapat digunakan seperlunya

Medan, 8 April 2016



(Iwan Fitrianto Rahmad, M.Kom )



# UNIVERSITAS POTENSI UTAMA

(FAKULTAS TEKNIK DAN ILMU KOMPUTER)

SK. Mendiknas R.I. No.: 103/D/O/2003

Kampus : Jl. K.L. Yos Sudarso KM 6,5 No. 3-A Telp. (061) 6640325 Fax:(061)6636830 Tanjung Mulia-Medan  
Website : <http://www.potensi-utama.ac.id>  
E-mail : [info@potensi-utama.ac.id](mailto:info@potensi-utama.ac.id)

## FORMULIR PENDAFTARAN SEMINAR HASIL SKRIPSI

### I. UMUM [Diisi oleh mahasiswa]

Nama Mahasiswa : MUHAMMAD NAZRI  
NIM : 111 0000 271  
Program Studi :  Teknik Industri (S1)  
 Teknik Informatika (S1)  
 Sistem Informasi (S1)

### II. PERSYARATAN SEMINAR HASIL SKRIPSI : [Diperiksa oleh Pembimbing]

- Sudah Melaksanakan Bimbingan dan Menyiapkan Laporan Skripsi (Rangkap 4):  
Pembimbing I :  Ya  Tidak  
Pembimbing II :  Ya  Tidak
- Sudah Melakukan Test Keberhasilan Program atau Alat Interface Sebagai Bahan Hasil Penelitian Skripsi:  
Pembimbing I :  Ya  Tidak  
Pembimbing II :  Ya  Tidak
- Sudah Ditanda Tangan Lembar Pengesahan Skripsi Sesuai dengan Format yang Diberikan:  
Pembimbing I :  Ya  Tidak  
Pembimbing II :  Ya  Tidak

[Ketentuan: Persyaratan harus dipenuhi]

### III. DATA SKRIPSI: [Diisi oleh mahasiswa]

- Judul :  
PERANCANGAN ROBOT PENJEMBAW MENGGUNAKAN SENSOR JARAK BERBASIS ANDROID
- Pembimbing I : EDY VICTOR HARYANTO, M.KOM
- Pembimbing II : IWAN FITRIANTO RAHMAD, M.KOM

Medan, .....

Mengetahui Ketua Prodi  
  
(Budi Triandi, M.Kom)

Pembimbing I

(Edy Victor Haryanto, M.Kom)

Pembimbing II

(Iwan Fitrianto Rahmad, M.Kom)

Diterima oleh Bagian Program Studi Tanggal : 5 Oktober 2016

(Sri Al-Fah)



**UNIVERSITAS POTENSI UTAMA**  
**(FAKULTAS TEKNIK DAN ILMU KOMPUTER)**  
**SK. Mendikbud No.: 424/E/O/2014**

Kampus : Jl. K.L. Yos Sudarso KM 6,5 No. 3-A Telp. (061) 6640525 Tanjung Mulia-Medan  
E-mail : [info@potensi-utama.ac.id](mailto:info@potensi-utama.ac.id)  
Website : <http://www.potensi-utama.ac.id>

**BERITA ACARA SEMINAR HASIL SKRIPSI**

Pada hari ini ..... tanggal ..... bulan ..... tahun ..... telah dilaksanakan seminar hasil Skripsi kepada :

**I. Data Mahasiswa**

NIM : 1110000271  
Nama : M. Nazri  
Tempat/Tgl. Lahir : Medan, 18 Oktober 1988  
Program Studi : Teknik Informatika  
Judul Skripsi : Perancangan Robot Penyeimbang Menggunakan Sensor Jarak Berbasis Android

**II. Dosen Pembimbing**

Pembimbing I		Pembimbing II	
NIDN	: 0130107701	NIDN	: 0125087903
Nama	: Edy Victor Haryanto, M.Kom	Nama	: Iwan Fitrianto Rahmad, M.Kom
Jabatan Akademik	: Lektor	Jabatan Akademik	: Lektor

**III. Team Pembanding**

Pembanding I  
NIDN : 0114097801  
Nama : Evri Ekadiansyah, M.Kom  
Jabatan Akademik : Lektor

No	Pembahasan BAB Skripsi	Keterangan
1		
2		- Sempurnakan perangkat.
3		
4		
5		
6		

Saran : .....

Demikian Berita Acara Seminar Hasil Skripsi ini diperbuat dengan sebenarnya untuk dipergunakan seperlunya.

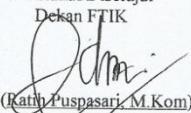
Penyaksi/Dosen Pembimbing

1. Edy Victor Haryanto, M.Kom
2. Iwan Fitrianto Rahmad, M.Kom

Medan, 15 Oktober 2016  
Team Pembanding

Pembanding I :  
Evri Ekadiansyah, M.Kom

Diketahui/Disetujui  
Dekan FTIK

  
(Katiy Puspasari, M.Kom)



**UNIVERSITAS POTENSI UTAMA**  
**(FAKULTAS TEKNIK DAN ILMU KOMPUTER)**  
**SK. Mendikbud No.: 424/E/O/2014**

Kampus : Jl. K.L. Yos Sudarso KM 6,5 No. 3-A Telp. (061) 6640525 Tanjung Mulia-Medan  
 E-mail : [info@potensi-utama.ac.id](mailto:info@potensi-utama.ac.id)  
 Website : <http://www.potensi-utama.ac.id>

**BERITA ACARA SEMINAR HASIL SKRIPSI**

Pada hari ini ..... tanggal ..... bulan ..... tahun ..... telah dilaksanakan seminar hasil Skripsi kepada :

**IV. Data Mahasiswa**

NIM : 1110000271  
 Nama : M. Nazri  
 Tempat/Tgl. Lahir : Medan, 18 Oktober 1988  
 Program Studi : Teknik Informatika  
 Judul Skripsi : Perancangan Robot Penyeimbang Menggunakan Sensor Jarak Berbasis Android

**V. Dosen Pembimbing**

<b>Pembimbing I</b>		<b>Pembimbing II</b>	
NIDN	: 0130107701	NIDN	: 0125087903
Nama	: Edy Victor Haryanto, M.Kom	Nama	: Iwan Fitrianto Rahmad, M.Kom
Jabatan Akademik	: Lektor	Jabatan Akademik	: Lektor

**VI. Team Pemanding**

**Pemanding II**  
 NIDN : 0018067506  
 Nama : Amirhud dalimunthe, ST, M.Kom  
 Jabatan Akademik : -

No	Pembahasan BAB Skripsi	Keterangan
1	Bab I	Flowchart dan metodologi penelitian.
2	Bab III	- Flowchart Perancangan Robot
3		- Pd Diagram blok → penjelasan buat tenaruf.
4		- Perancangan Mehand → semua yg dibuat
5	Bab IV	Metode Pengujian.
6		

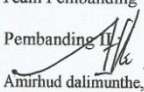
Saran : .....

Demikian Berita Acara Seminar Hasil Skripsi ini diperbuat dengan sebenarnya untuk dipergunakan seperlunya.

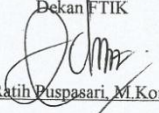
Penyaksi/Dosen Pembimbing

1. Edy Victor Haryanto, M.Kom
2. Iwan Fitrianto Rahmad, M.Kom

Medan, 15 Oktober 2016  
 Team Pemanding

Pemanding II  
  
 Amirhud dalimunthe, ST, M.Kom

Diketahui/Disetujui  
 Dekan FTIK

  
 (Ratih Puspasari, M.Kom)



**DOKUMEN LEVEL  
FORM**

**NO. DOKUMEN**  
F-FTIK-13-08

**JUDUL.**  
DAFTAR HADIR MAHASISWA PESERTA SEMINAR HASIL SKRIPSI

**AREA**  
PROGRAM STUDI

Tanggal Terbit : 07 Nov 2014  
Tanggal Efektif : 14 Nov 2014  
Halaman : 1 dari 1

**NO.REVISI**  
00

**PROGRAM STUDI : TEKNIK INFORMATIKA**

JAM	NAMA DOSEN	TANDA TANGAN				NAMA MAHASISWA	TANDA TANGAN
		Pembimbing I	Pembimbing II	Pembanding I	Pembanding II		
10.00 - 11.00 WIB	Edy Victor Hartanto, M.Kom						
10.00 - 11.00 WIB	Iwan Fitriono Rahmad, M.Kom						
10.00 - 11.00 WIB	Evi Ekadiansyah, M.Kom					M. Nazri	
10.00 - 11.00 WIB	Amirhuda dalimunde, ST, M.Kom						

Medan, 15 Oktober 2016  
Ketua Program Teknik Informatika

Budi Triandi, M.Kom



# UNIVERSITAS POTENSI UTAMA (FAKULTAS TEKNIK DAN ILMU KOMPUTER)

SK. Mendikbud No.: 424/E/O/2014

Kampus : Jl. K.L. Yos Sudarso KM 6,5 No. 3-A Telp. (061) 6640525 Tanjung Mulia-Medan  
E-mail : info@potensi-utama.ac.id  
Website : http://www.potensi-utama.ac.id

## FORMULIR PENDAFTARAN UJIAN SIDANG SKRIPSI

### I. UMUM [Diisi oleh mahasiswa]

Nama Mahasiswa : MUHAMMAD NAZRI  
NIM : 111 0000 271  
Program Studi :  Teknik Industri (S1)  
 Teknik Informatika (S1)  
 Sistem Informasi (S1)

### II. PERSYARATAN UJIAN SIDANG SKRIPSI : [Diperiksa oleh Pembanding]

- Sudah Melaksanakan Seminar Proposal Skripsi :  
Pembimbing I :  Ya  Tidak  
Pembimbing II :  Ya  Tidak
- Sudah Melaksanakan Bimbingan dan Menyiapkan Laporan Tugas Akhir (Rangkap 3).  
Pembimbing I :  Ya  Tidak  
Pembimbing II :  Ya  Tidak
- Sudah Ditanda Tanganinya Lembar Persetujuan Sidang Skripsi Oleh Pembimbing Sesuai dengan Format yang Diberikan :  
Pembimbing I :  Ya  Tidak  
Pembimbing II :  Ya  Tidak

[Ketentuan: Persyaratan harus dipenuhi]

### III. DATA SKRIPSI: [Diisi oleh mahasiswa]

- Judul :  
PERANCANGAN ROBOT PENYEIMBANG MENGGUNAKAN SENSOR JARAK BERBASIS ANDROID
- Pembimbing I : EDY VICTOR HARYANTO, M.Kom
- Pembimbing II : IWAN FITRIANTO RAHMAD, M.Kom
- Pembanding I : EURI EKADIANSYAH, M.Kom
- Pembanding II : AMIRHUD DALIMUNTHE, ST, M.Kom

Medan, 10-11-2016

Mengajar Ketua Prodi  
  
(Budi Triandji, M.Kom)

Pembanding I  
  
(Euri EkadianSyah, M.Kom)

Pembanding II  
  
(Amirhud Dalimunthe, ST, M.Kom)

Diterima oleh Bagian BAAK Tanggal : 10-11-2016

(.....)