



**RANCANG BANGUN SISTEM NAVIGASI ROBOT BERODA UNTUK  
MONITORING SUHU RUANGAN MENGGUNAKAN METODE  
BEHAVIOR BASED DENGAN DISPLAY ANDROID**

**Skripsi**

Oleh

**Ikhsanudin**

**NIM 141910201032**

**JURUSAN TEKNIK ELEKTRO STRATA 1  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2018**



**RANCANG BANGUN SISTEM NAVIGASI ROBOT BERODA UNTUK  
MONITORING SUHU RUANGAN MENGGUNAKAN METODE  
BEHAVIOR BASED DENGAN DISPLAY ANDROID**

**Skripsi**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan Program Studi Elektro (S1)  
dan mencapai gelar Sarjana Teknik

Oleh

**Ikhsanudin**

**NIM 141910201032**

**JURUSAN TEKNIK ELEKTRO STRATA 1  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2018**

## **PERSEMBAHAN**

Alhamdulillah, puji syukur kepada Allah SWT. Diri ini tiada daya tanpa kekuatan dari-Mu. Shalawat dan salamku kepada suri tauladanku Nabi Muhammad SAW. Ku harap syafa'atmu di penghujung hari nanti. Dengan segala ketulusan hati kupersembahkan laporan skripsi ini kepada orang-orang yang mempunyai ketulusan jiwa yang senantiasa membimbingku dan menjadi sahabat selama aku dilahirkan ke dunia ini.

### ***Yang Pertama***

Ayah dan Ibunda tersayang. Engkaulah Pelita hatiku yang telah mengasihiku dan menyayangiku dari lahir sampai mengerti luasnya ilmu di dunia ini dan sesuci do'a malam hari. Terima kasih atas semua yang telah engkau berikan kepadaku.

### ***Yang Kedua***

Kedua kakakku tercinta yang selalu memberikan dukungan serta cinta kasihnya kepadaku.

### ***Yang Ketiga***

Semua dosen-dosenku yang telah memberikan bimbingan dan ilmu yang tidak bisa kuhitung berapa banyaknya barakah dan do'anya.

### ***Yang Terakhir***

Semua sahabatku khususnya KETEK UJ 14 seperjuangan di bumi Universitas Jember, kuatkan tekadmu tuk hadapi rintangan, karena sesungguhnya Allah bersama kita.

**MOTTO**

*“Jangan malu terlihat bodoh malulah saat kita pura – pura pintar”  
(Albert Einstein)*

*“Manusia itu penuh kekurangan, karena itu kita butuh teman yang saling  
menolong untuk menutupi kekurangan masing-masing”  
(Kosuke Ueki – The Law of Ueki)*

*“Merenung dan belajar dari kesalahan adalah bagian dari kehidupan”  
(All Might – Boku No Hero Academy)*

*“Tidak ada tombol reset dalam kehidupan, tapi setidaknya kamu bisa me-reset  
hubunganmu”  
(Hakigara Hachiman - Oregairu)*

*“When your friend is having a hard time, you should be there for him”  
(Usopp – One Piece)*

**PERNYATAAN**

Saya yang bertanda tangan dibawah ini:

Nama : Ikhsanudin

NIM : 141910201032

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Rancang Bangun Sistem Navigasi Robot Beroda Untuk *Monitoring* Suhu Ruang Menggunakan Metode *Behavior Based* Dengan Display Android" adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun dan bukan karya jiplakan. Saya bertanggung jawab penuh atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Juni 2018

Yang menyatakan,

Ikhsanudin

NIM 141910201032

**SKRIPSI**

**RANCANG BANGUN SISTEM NAVIGASI ROBOT BERODA UNTUK  
MONITORING SUHU RUANGAN MENGGUNAKAN METODE  
BEHAVIOR BASED DENGAN DISPLAY ANDROID**

Oleh

**Ikhsanudin**

**NIM 141910201032**

Pembimbing :

Dosen Pembimbing Utama : **Khairul Anam, S.T., M.T.,Ph.D**

Dosen Pembimbing Anggota : **Sumardi, S.T., M.T.**

**PENGESAHAN**

Skripsi berjudul “**Rancang Bangun Sistem Navigasi Robot Beroda Untuk Monitoring Suhu Ruangan Menggunakan Metode *Behavior Based* Dengan Display Android**” karya Ikhsanudin telah diuji dan disahkan pada :

Hari : Jum'at

Tanggal : 8 Juni 2018

Tempat : Fakultas Teknik Universitas Jember

Tim penguji,

Ketua,

Sekretaris,

**Khairul Anam, S.T., M.T., Ph.D**

NIP 197804052005011002

**Sumardi, S.T., M.T.**

NIP 196701131998021001

Anggota I,

Anggota II,

**Catur Suko Sarwono S.T., M.Si.**

NIP 196801191997021001

**Dodi Setiabudi S.T., M.T.**

NIP 198405312008121004

Mengesahkan  
Dekan,

**Dr. Ir. Entin Hidayah, M.U.M.**

NIP 196612151995032001

Rancang Bangun Sistem Navigasi Robot Beroda Dengan Menggunakan Metode  
*Behavior Based* Untuk *Monitoring* Suhu Ruangan Dengan *Display* Android

**IKHSANUDIN**

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

## **ABSTRAK**

Robot adalah sebuah alat mekanik dan elektrik yang dapat melakukan tugas fisik, dengan atau tanpa pengawasan dari manusia. *Autonomus mobile robot* adalah robot yang mampu berperilaku secara mandiri (hanya sesekali membutuhkan perintah). Semakin berkembangnya ilmu robot berdampak pada kehidupan dimana robot dijadikan pengganti manusia dalam mengerjakan pekerjaan yang sulit dikerjakan manusia. Salah satunya yaitu ketika terjadi bencana alam kenaikan suhu ekstrim dimana manusia tidak dapat terjun langsung untuk mengatasinya sehingga diperlukanlah robot yang mampu mendeteksi suhu panas tersebut dan mampu mentransfer data secara otomatis. Untuk membuat robot mampu berjalan otomatis diperlukan metode *behavior based* sebagai kontrol dari robot serta kontrol PID sebagai kontrol motor pada aktuator robot.

**Kata kunci:** *Autonomus mobile robot*, *behavior based*, PID.



*(Design Of Wheeled Robot Navigation System For Room Temperature Monitoring Using Behavior Based Method With Android Display)*

**IKHSANUDIN**

*Department of Electrical Engineering, Faculty of Engineering, University of Jember*

## **ABSTRACT**

*Robot is a mechanical and electrical device that can perform physical tasks, with or without human control. Autonomus mobile robot is a robot that capable of behaving independently (only once need direction). The increase of the development of robot science influence to human life and substitute in finishing and difficult work. Such as if there is a natural disasters like increasing of the extreame temperatures where human can't solve it directly, so they need robot to detect its temperature and capable to transfer the data automatically. To make a robot can be run automatically, we need behavior based method as a robot control and PID motor control on its robot actuator.*

**Keywords:** *Autonomus mobile robot, behavior based, PID*

## RINGKASAN

**Rancang Bangun Sistem Navigasi Robot Beroda Untuk *Monitoring* Suhu Ruang Menggunakan Metode *Behavior Based* Dengan Display Android;** Ikhsanudin; 141910201032; 2018; 76 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Robot adalah sebuah alat mekanik dan elektrik yang dapat melakukan tugas fisik, dengan atau tanpa pengawasan dari manusia. Robot biasa digunakan untuk membantu pekerjaan manusia dimana pekerjaan itu bersifat berat, berbahaya, pekerjaan yang memiliki kecepatan yang berulang dan kotor, semakin berkembangnya ilmu robot juga berdampak di kehidupan sehari – hari dimana manusia dapat dimudahkan dalam melakukan pekerjaannya atau dapat digunakan sebagai pengganti manusia diantaranya ketika terjadi bencana alam ketika manusia tidak dapat memasuki sebuah ruangan atau kawasan karena bencana robot dapat mewakilinya salah satunya yaitu bencana kenaikan suhu yang ekstrim dimana suhu berubah secara drastis di sebuah kawasan atau ruangan yang menyebabkan manusia tidak mungkin untuk memasukinya.

Dalam inovasi itu sendiri telah sampai pada titik dimana robot dapat berjalan sendiri (*autonomus mobile robot*). *Autonomus mobile robot* adalah robot yang mampu berperilaku secara mandiri (hanya sesekali membutuhkan perintah). Untuk itu dibuatlah kontrol *behavior based* untuk membantu koordinasi setiap perilaku yang diprogram sehingga membuat robot mampu melaksanakan setiap tugas yang diberikan secara baik. Kontrol *behavior based* dapat lebih dilengkapi lagi dengan kontrol PID dimana kontrol PID disini membantu dalam proses respon motor yang digunakan, sehingga pergerakan dari motor itu sendiri lebih responsif.

Dalam pengujian respon motor menggunakan kontrol PID digunakan beberapa kondisi robot. Pengujian kontrol PID untuk respon motor pada robot sendiri dilakukan beberapa tahap dengan cara *trial and error* diantaranya kontrol P, kontrol PI kontrol PD dan Kontrol PID. Selain dengan cara *trial and error* juga

dilakukan pengujian PID *Ziegler-Nichols* tipe 2 yang kemudian dibandingkan dengan kontrol PID *trial and error*. Hasil pengujian kontrol PID secara *trial and error* lebih baik dibandingkan dengan kontrol PID *Ziegler-Nichols* tipe 2.

Robot beroda pemantau suhu ruangan ini sendiri untuk proses penyimpanan data suhunya dihubungkan dengan sebuah web yaitu thingspeak.com dimana pada web ini memberi fasilitas yang banyak salah satunya yaitu dapat membuat *channel* untuk menyimpan sebuah data berupa grafik dan mampu dikonversi menjadi data excel. Serta web ini dapat dikoneksikan langsung dengan aplikasi yang ada pada android sehingga memudahkan dalam perancangan *software* pada android.

Pada proses navigasi robot beroda pada sebuah ruangan yang sebenarnya dimana robot melakukan proses navigasi sekaligus monitoring suhu ruangan, dimana dalam proses pengujian robot itu sendiri robot masih melakukan proses navigasi yang belum sempurna dimana saat robot bernavigasi di dalam ruangan terjadi beberapa kali tabrakan serta kehilangan arah pada robot, dimana proses navigasi robot didalam ruangan sebenarnya itu sendiri memiliki 60% tingkat keberhasilan saat proses percobaan. Untuk proses pengiriman data suhu yang telah terekam sendiri dilakukan setiap 30 detik sekali dan dapat langsung dilihat perubahan dari suhu yang terekam melalui aplikasi yang telah dibuat pada android.

## PRAKATA

Puji syukur kehadiran Allah SWT yang maha kuasa atas segalanya, karena dengan ridho, hidayah dan petunjukNya, penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Sistem Navigasi Robot Beroda Untuk *Monitoring* Suhu Ruangan Menggunakan Metode *Behavior Based* Dengan Display Android”. Selama penyusunan skripsi ini penulis mendapat bantuan berbagai pihak yang turut memberikan bantuan berupa motivasi, inspirasi, bimbingan, doa, fasilitas dan dukungan lainnya yang membantu memperlancar pengerjaan skripsi ini.

Untuk itu penulis mengucapkan terimakasih kepada.

1. Allah SWT yang telah melimpahkan rezeki, rahmat, hidayah dan karunia serta kasih sayang-Nya sehingga penulis mampu menyelesaikan skripsi ini.
2. Nabi besar Muhammad SAW, yang telah menjadi suri tauladan bagi seluruh umat.
3. Ibu Dr. Ir. Entin Hidayah, M.U.M., selaku Dekan Fakultas Teknik Universitas Jember
4. Bapak Dr. Bambang Sri Kaloko, S.T., M.T., Selaku Ketua Jurusan Teknik Elektro Universitas Jember;
5. Bapak Khairul Anam, S.T., M.T., Ph.D dan Bapak Sumardi, S.T., M.T. selaku dosen pembimbing yang telah membimbing menyelesaikan tugas akhir ini;
6. Bapak Catur Suko Sarwono S.T., M.Si. dan Bapak Dodi Setiabudi S.T., M.T. selaku dosen penguji yang sudah memberikan saran untuk memperbaiki tugas akhir ini;
7. Ayahanda Slamet dan ibunda Tumini telah menjadi orang tua terhebat, yang selalu memberikan motivasi, nasehat, cinta, perhatian, dan kasih sayang serta doa yang tentu takkan bisa terbalaskan;
8. Kedua kakak tersayang Erna Susanti dan Zainul Arifin senantiasa memberikan doa dan bantuan baik berupa moril maupun materi selama penyusunan Tugas Akhir ini;

9. Sahabat terdekatku Neny Indah Luvita yang selalu membantu, menemani dan mendampingi dalam pengerjaan Tugas Akhir ini tanpa lelah;
10. Sahabat-sahabat seperjuangan Teknik Elektro 2014 yang telah memberikan motivasi dan semangat dalam perjuangan di bangku kuliah;
11. semua pihak yang tidak dapat disebutkan satu persatu.

Semoga skripsi ini dapat bermanfaat dalam mengembangkan ilmu pengetahuan khususnya untuk disiplin ilmu teknik elektro. Kritik dan saran yang membangun diharapkan terus mengalir untuk lebih menyempurnakan skripsi ini dan dapat dikembangkan untuk penelitian selanjutnya;

Jember, 8 Juni 2018

Penulis

DAFTAR ISI

HALAMAN JUDUL .....	i
PERSEMBAHAN .....	ii
MOTTO .....	iii
PERNYATAN.....	iv
HALAMAN PEMBIMBING.....	v
PENGESAHAN .....	vi
ABSTRAK .....	vi
RINGKASAN .....	ix
PRAKATA .....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR .....	xvi
DAFTAR TABEL .....	xx
DAFTAR LAMPIRAN .....	xxi
<b>BAB 1. PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.4 Tujuan.....	2
1.5 Batsan Maslah .....	3
<b>BAB 2. TINJAUAN PUSTAKA .....</b>	<b>4</b>
2.1 Robot Beroda .....	4
2.2 Arduino Mega 2560 .....	5
2.3 LCD .....	7
2.4 <i>Driver Motor</i> .....	8
2.5 Motor DC .....	9
2.6 <i>Sensor Rotary Encoder</i> .....	10
2.7 <i>Sensor Suhu DHT 11</i> .....	10
2.8 <i>Esp 8266</i> .....	11
2.9 <i>Buzzer</i> .....	12

<b>2.10 Sistem Kendali PID (<i>Proportional Integral Derivative</i>)</b> .....	<b>12</b>
2.10.1 Cara Mentuning Manual Parameter PID .....	13
2.10.2 Metoda Ziegler-Nichols .....	14
<b>BAB 3. METODOLOGI PENELITIAN</b> .....	<b>17</b>
<b>3.1 Waktu dan Tempat</b> .....	<b>17</b>
3.1.1 Tempat .....	17
3.1.2 Waktu .....	17
<b>3.2 Alat dan Bahan</b> .....	<b>17</b>
<b>3.3 Metodologi Pelaksanaan dan Penelitian</b> .....	<b>19</b>
<b>3.4 Perancangan Perangkat Keras</b> .....	<b>20</b>
3.4.1 Desain Robot Beroda .....	20
3.4.2 Desain Elektronik .....	22
<b>3.5 Sistem Kendali Robot <i>Behavior Based Control</i></b> .....	<b>23</b>
3.5.1 <i>Behavior Based Control</i> .....	23
3.5.2 Bagian <i>Behavior Based Control</i> .....	24
3.5.2.1 <i>Wander</i> .....	24
3.5.2.2 Menghindari halangan & menyusur ruangan .....	24
3.5.2.3 Mendeteksi Suhu Panas .....	25
<b>3.6 Sistem Kontrol Robot Menggunakan PID</b> .....	<b>25</b>
<b>3.7 Flowchart</b> .....	<b>26</b>
<b>3.8 Metode Perancangan Pengujian</b> .....	<b>28</b>
3.8.1 Perancangan Pengujian Motor DC .....	28
3.8.2 Perancangan Pengujian Motor esp 8266 .....	28
3.8.3 Perancangan Pengujian Kontrol PID .....	29
3.8.4 Perancangan Pengujian <i>Behavior Based Control</i> .....	29
3.8.5 Perancangan Pengujian Keseluruhan .....	29
<b>3.8 Hasil Perancangan Alat</b> .....	<b>30</b>
<b>3.9 Hasil Perancangan Aplikasi Pada Android</b> .....	<b>30</b>
<b>BAB 4. HASIL DAN PEMBAHASAN</b> .....	<b>36</b>
<b>4.1 Sensor Jarak HCSR-04</b> .....	<b>36</b>
4.1.1 Kalibrasi Sensor Jarak HCSR-04 .....	36

4.1.2 Pengujian Sensor Jarak HCSR-04 .....	38
<b>4.2 Sensor Rotary Encoder .....</b>	<b>39</b>
4.2.1 Kalibrasi Sensor <i>Rotary Encoder</i> .....	39
4.2.2 Pengujian Sensor <i>Rotary Encoder</i> .....	40
<b>4.3 Sensor Suhu DHT 11 .....</b>	<b>41</b>
4.2.1 Kalibrasi Sensor Suhu DHT 11 .....	41
4.2.2 Pengujian Sensor Suhu DHT 11 .....	42
<b>4.4 Kontrol PID .....</b>	<b>43</b>
4.4.1 <i>Tunning</i> Parameter Kontrol PID Manual .....	43
4.4.2 <i>Tunning</i> Kontrol PID <i>Ziegler-Nichols</i> Tipe 2 .....	45
4.4.3 Pengujian Kontrol PID .....	48
4.4.3.1 Pengujian Kontrol PID Tanpa Beban .....	48
4.4.3.2 Pengujian Kontrol PID Beban .....	51
4.4.3.2.1 Pengujian Kontrol PID Beban 200 g .....	51
4.4.3.2.2 Pengujian Kontrol PID Beban 320 g .....	53
4.4.3.3 Pengujian Kontrol Pada Lantai Berpasir .....	56
<b>4.5 Pengujian ESP 8266 .....</b>	<b>59</b>
<b>4.6 Pengujian Metode <i>Behavior Based Control</i> .....</b>	<b>61</b>
<b>4.7 Pengujian Keseluruhan .....</b>	<b>71</b>
4.7.1 Pengujian Keseluruhan <i>Behavior</i> .....	71
4.7.2 Pengujian Navigasi Robot Pada Ruang 1 & 2 .....	74
<b>BAB 5. KESIMPULAN DAN SARAN .....</b>	<b>77</b>
5.1 Kesimpulan .....	77
5.2 Saran .....	77
<b>DAFTAR PUSTAKA .....</b>	<b>78</b>
<b>LAMPIRAN .....</b>	<b>79</b>



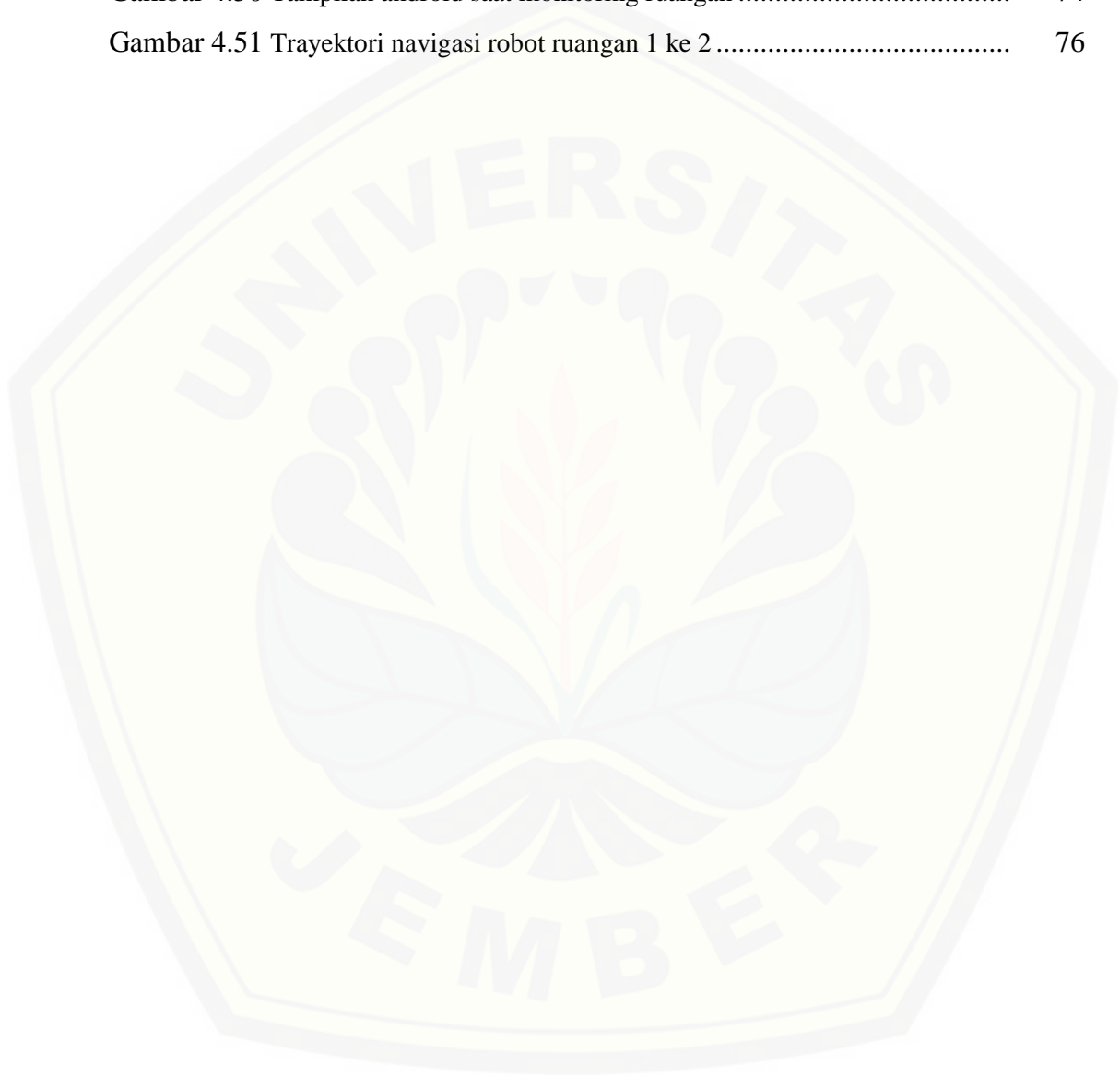
**DAFTAR GAMBAR**

Gambar 2.1 Pengendalian <i>ackreman steering</i> .....	4
Gambar 2.2 Pengendalian <i>differential sttering</i> .....	5
Gambar 2.3 <i>Board</i> Arduino Mega 2560 .....	6
Gambar 2.4 LCD .....	7
Gambar 2.5 Modul <i>driver motor</i> .....	8
Gambar 2.6 Konfigurasi PIN <i>Driver Motor</i> L298 .....	9
Gambar 2.7 Motor DC .....	9
Gambar 2.8 <i>Optical Rotary Encoder</i> .....	10
Gambar 2.9 Sensor Suhu DHT 11 .....	11
Gambar 2.10 ESP 8266-01 .....	11
Gambar 2.11 Tampilan <i>thingspeak.com</i> .....	12
Gambar 2.12 <i>Buzzer</i> .....	12
Gambar 2.13 Blok diagram Ppengendali PID.....	13
Gambar 2.14 Sistem <i>Ziegler-Nichols</i> tipe 1 .....	14
Gambar 2.15 Proses desain penentuan parameter L dan T .....	15
Gambar 2.16 Sistem <i>closed loop</i> dengan menggunakan Kp .....	15
Gambar 2.17 Proses mendesain menentukan parameter Pcr .....	16
Gambar 3.1 <i>Flowchart</i> penelitian .....	19
Gambar 3.2 Desain robot tampak atas .....	20
Gambar 3.3 Desain robot tampak kiri atas .....	21
Gambar 3.4 Desain robot tampak depan.....	21
Gambar 3.5 Desain elektronik .....	22
Gambar 3.6 Diagram blok <i>Behavior based control</i> .....	23
Gambar 3.7 Diagram blok <i>wander</i> .....	24
Gambar 3.8 Diagram blok menghindari halangan dan menyusur ruangan .....	24
Gambar 3.9 Diagram blok menemukan suhu panas.....	25
Gambar 3.10 Diagram blok kontrol PID.....	25
Gambar 3.11 <i>Flowchart</i> keseluruhan.....	27

Gambar 3.12 <i>Flowchart</i> menyusur tembok .....	28
Gambar 3.13 Robot tampak atas .....	30
Gambar 3.14 Robot tampak samping.....	30
Gambar 3.15 Robot tampak depan.....	30
Gambar 3.16 Tampilan awal web .....	31
Gambar 3.17 Tampilan awal setelah masuk pada web .....	31
Gambar 3.18 Tampilan pada <i>channel</i> .....	31
Gambar 3.19 Tampilan akhir <i>channels</i> .....	32
Gambar 3.20 Tampilan awal virtuino .....	32
Gambar 3.21 Tampilan awal pengaturan sambungan .....	33
Gambar 3.22 Tampilan saat pengaturan sambungan .....	33
Gambar 3.23 Tampilan ketika aplikasi sudah sambungan.....	33
Gambar 3.24 Menu awal pembuatan tampilan.....	34
Gambar 3.25 Tampilan pengaturan untuk tampilan <i>text</i> .....	34
Gambar 3.26 Tampilan pengaturan untuk tampilan angka .....	34
Gambar 3.27 Tampilan pengaturan untuk tampilan grafik .....	35
Gambar 3.28 Hasil akhir tampilan android.....	35
Gambar 4.1 Grafik kalibrasi sensor jarak .....	37
Gambar 4.2 Pengujian sensor jarak .....	39
Gambar 4.3 Grafik kalibrasi sensor <i>rotary encoder</i> .....	40
Gambar 4.4 Grafik pengujian sensor <i>rotary encoder</i> .....	41
Gambar 4.5 Grafik kalibrasi sensor suhu .....	42
Gambar 4.6 Grafik pengujian sensor suhu.....	42
Gambar 4.7 Grafik respon kontrol <i>proporsional</i> .....	43
Gambar 4.8 Grafik reespon kontrol <i>proporsional</i> dan <i>derivatif</i> .....	44
Gambar 4.9 Grafik respon kontrol <i>proporsional</i> , intergal dan <i>derivatif</i> .....	45
Gambar 4.10 Grafik kontrol <i>proporsional</i> untuk menentukan Pcr.....	46
Gambar 4.11 Grafik respon kontrol PID <i>Ziegler-Nichols</i> tipe 2.....	47
Gambar 4.12 Grafik respon kontrol PID dengan <i>setpoint</i> 50 .....	48
Gambar 4.13 Grafik respon kontrol PID dengan <i>setpoint</i> 150 .....	49
Gambar 4.14 Grafik respon kontrol PID dengan <i>setpoint</i> 200.....	49

Gambar 4.15 Grafik respon kontrol PID dengan <i>setpoint</i> 250 .....	50
Gambar 4.16 Grafik respon kontrol PID dengan <i>setpoint</i> berubah .....	50
Gambar 4.17 Grafik respon kontrol PID beban 200 g dengan <i>setpoint</i> 100....	51
Gambar 4.18 Grafik respon kontrol PID beban 200 g dengan <i>setpoint</i> 150....	51
Gambar 4.19 Grafik respon kontrol PID beban 200 g dengan <i>setpoint</i> 200....	52
Gambar 4.20 Grafik respon PID beban 200 g dengan <i>setpoint</i> berubah.....	52
Gambar 4.21 Pengujian kontrol PID dengan beban 200 g.....	53
Gambar 4.22 Grafik respon kontrol PID beban 320 g dengan <i>setpoint</i> 100 .....	53
Gambar 4.23 Grafik respon kontrol PID beban 320 g dengan <i>setpoint</i> 150 .....	54
Gambar 4.24 Grafik respon kontrol PID beban 320 g dengan <i>setpoint</i> 200 .....	54
Gambar 4.25 Grafik respon PID beban 320 g dengan <i>setpoint</i> berubah.....	55
Gambar 4.26 Pengujian kontrol PID dengan beban 320 g.....	55
Gambar 4.27 Grafik respon PID pada rantai berpasir dengan <i>setpoint</i> .....	56
Gambar 4.28 Grafik respon PID pada rantai berpasir dengan <i>setpoint</i> 100.....	56
Gambar 4.29 Grafik respon PID pada rantai berpasir dengan <i>setpoint</i> 150.....	57
Gambar 4.30 Grafik respon PID pada rantai berpasir dengan <i>setpoint</i> 200.....	57
Gambar 4.31 Grafik respon PID pada rantai berpasir dengan <i>setpoint</i> berubah	58
Gambar 4.32 Pengujian kontrol PID dengan rantai berpasir.....	58
Gambar 4.33 Pengujian LOS esp 8266 .....	59
Gambar 4.34 Pengujian NLOS esp 8266 .....	60
Gambar 4.35 Robot ditengah ruangan .....	62
Gambar 4.36 Gambar trayektori robot <i>wander</i> .....	63
Gambar 4.37 Robot berada diposisi untuk berjalan lurus .....	63
Gambar 4.38 Gambar trayektori robot berjalan lurus .....	64
Gambar 4.39 Robot berada diposisi untuk serong kiri .....	64
Gambar 4.40 Gambar trayektori robot serong kiri.....	65
Gambar 4.41 Robot berada diposisi untuk belok kiri .....	65
Gambar 4.42 Gambar trayektori robot belok kiri .....	66
Gambar 4.43 Robot berada diposisi untuk serong kanan .....	66
Gambar 4.44 Gambar trayektori robot serong kanan .....	67
Gambar 4.45 Robot berada diposisi untuk belok kanan .....	67

Gambar 4.46 Gambar trayektori robot belok kanan .....	68
Gambar 4.47 Robot mendeteksi suhu panas .....	69
Gambar 4.48 Robot memulai pengujian keseluruhan .....	71
Gambar 4.49 Gambar trayektori pengujian keseluruhan .....	72
Gambar 4.50 Tampilan android saat monitoring ruangan .....	74
Gambar 4.51 Trayektori navigasi robot ruangan 1 ke 2 .....	76



**DAFTAR TABEL**

Tabel 2.1 Spesifikasi arduino mega 2560 .....	6
Tabel 2.2 Parameter PID .....	14
Tabel 2.3 Parameter PID untuk ZN tipe 1 .....	15
Tabel 2.4 Parameter PID Untuk ZN Tipe 2 .....	16
Tabel 3.1 Daftar Peralatan yang Digunakan .....	18
Tabel 4.1 Kalibrasi sensor jarak.....	37
Tabel 4.2 Pengujian sensor jarak .....	38
Tabel 4.3 Kalibrasi sensor <i>rotary encoder</i> .....	39
Tabel 4.4 Pengujian sensor <i>rotary encoder</i> .....	40
Tabel 4.5 Kalibrasi sensor suhu DHT 11 .....	41
Tabel 4.6 Pengujian sensor suhu DHT 11.....	42
Tabel 4.7 Respon kontrol <i>proporsional</i> .....	43
Tabel 4.8 Respon kontrol <i>proporsional</i> dan <i>derivatif</i> .....	44
Tabel 4.9 Respon kontrol <i>proporsional</i> , intergal dan <i>derivatif</i> .....	44
Tabel 4.10 Perbandingan kontrol PID <i>tuning</i> manual dan <i>Ziegler-Nichols</i> ...	47
Tabel 4.11 Pengujian koneksi esp8266 saat LOS .....	59
Tabel 4.12 Pengujian koneksi esp8266 saat NLOS .....	60
Tabel 4.13 Pengujian metode <i>behavior based</i> .....	61
Tabel 4.14 Data pengujian menghindari halangan dan menyusur tembok .....	68
Tabel 4.15 Data pengujian mendeteksi suhu panas .....	69
Tabel 4.16 Data pengujian robot monitoring ruangan .....	72
Tabel 4.17 Data suhu yang direkam oleh robot .....	73
Tabel 4.18 Data Pengujian navigasi dari ruangan 1 dan 2 .....	74

**DAFTAR LAMPIRAN**

A. <i>Listing</i> Program Arduino Mega 2560 .....	78
B. Gambar <i>Hardware</i> .....	85
C. Gambar <i>Software</i> Aplikasi Android .....	86



## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Semakin berkembangnya ilmu pengetahuan dan teknologi robotika saat ini telah membuat banyak orang serta negara melakukan sebuah penelitian untuk terus membuat inovasi dari robot. Salah satunya yang paling banyak diinovasi serta diteliti adalah robot beroda, robot beroda sendiri digemari karena kecepatannya dalam melintasi bidang yang rata dan kemudahan dalam hal mendesain. Dalam inovasi itu sendiri telah sampai pada titik dimana robot dapat berjalan sendiri (*autonomus robot*). *Autonomus robot* adalah robot yang mampu berperilaku secara mandiri (hanya sesekali membutuhkan perintah). Saat ini perkembangan teknologi dan penelitian lebih fokus kearah *autonomus robot*, yakni robot yang mampu berpindah posisi secara mandiri.

Dalam *autonomous mobile robot* hal yang paling mendasar adalah robot mampu mengenali objek-objek yang ada disekitarnya kemudian hal yang paling penting yaitu sistem navigasi robot. Sistem navigasi pada *autonomous mobile robot* digunakan agar robot mampu bergerak secara mandiri dan mempunyai kemampuan untuk memandu pergerakan dari suatu posisi ke posisi lain yang dituju melalui penentuan posisi dan arah geraknya.

Dalam proses navigasi robot itu sensor dibutuhkan sensor yang digunakan sebagai masukan untuk sistem navigasi robot diantaranya *sensor rotary encoder* dan sensor ultrasonik sebagai sistem navigasi robot agar dapat menuju .(Kariyanto dkk., 2011)

Semakin berkembangnya ilmu robot juga berdampak dikehidupan sehari – hari dimana manusia dapat dimudahkan dalam melakukan pekerjaannya atau dapat digunakan sebagai pengganti manusia diantaranya ketika terjadi bencana alam ketika manusia tidak dapat memasuki sebuah ruangan atau kawasan karena bencana robot dapat mewakilinya salah satunya yaitu bencana kenaikan suhu yang ekstrim dimana suhu berubah secara drastis disebuah kawasan atau ruangan yang menyebabkan manusia tidak mungkin untuk memasukinya untuk itu digunakanlah robot beroda yang dapat menggantikan pekerjaan manusia. Sehingga semakin

banyak pengembangan mengenai kontrol robot beroda. Hal ini juga didukung oleh banyaknya penelitian tentang sistem navigasi robot yang berguna untuk membantu pekerjaan manusia salah satunya adalah Muhamad Khoiril Rijal, dkk (2016). Membuat penelitian tentang “Robot Beroda Pemantau Suhu Dengan *Display* pada LCD Berbasis Arduino Mega 2560” robot ini digunakan untuk memantau suhu ruangan menggunakan sensor LM35. (Rijal, 2016)

Berdasarkan hal yang disajikan pada penelitian diatas dimana robot masih melakukan navigasi dengan dikontrol menggunakan *joy stick* dan masih melakukan proses navigasi pada ruangan buatan sehingga penulis melakukan inovasi dengan membuat suatu “Rancang Bangun Sistem Navigasi Robot Beroda Menggunakan Metode *Behavior Based* Untuk *Monitoring* Suhu Ruangan Dengan *Display* Android”.

## 1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini antara lain:

1. Bagaimana cara membuat rancang bangun robot navigasi pada sebuah ruangan untuk proses *monitoring* suhu ?
2. Bagaimana cara merancang kontrol *behavior based* dan kontrol PID pada robot beroda untuk melakukan proses navigasi ?
3. Bagaimana cara menguji robot beroda pada ruangan yang sebenarnya ?

## 1.3 Tujuan

Tujuan dari penelitian ini antara lain:

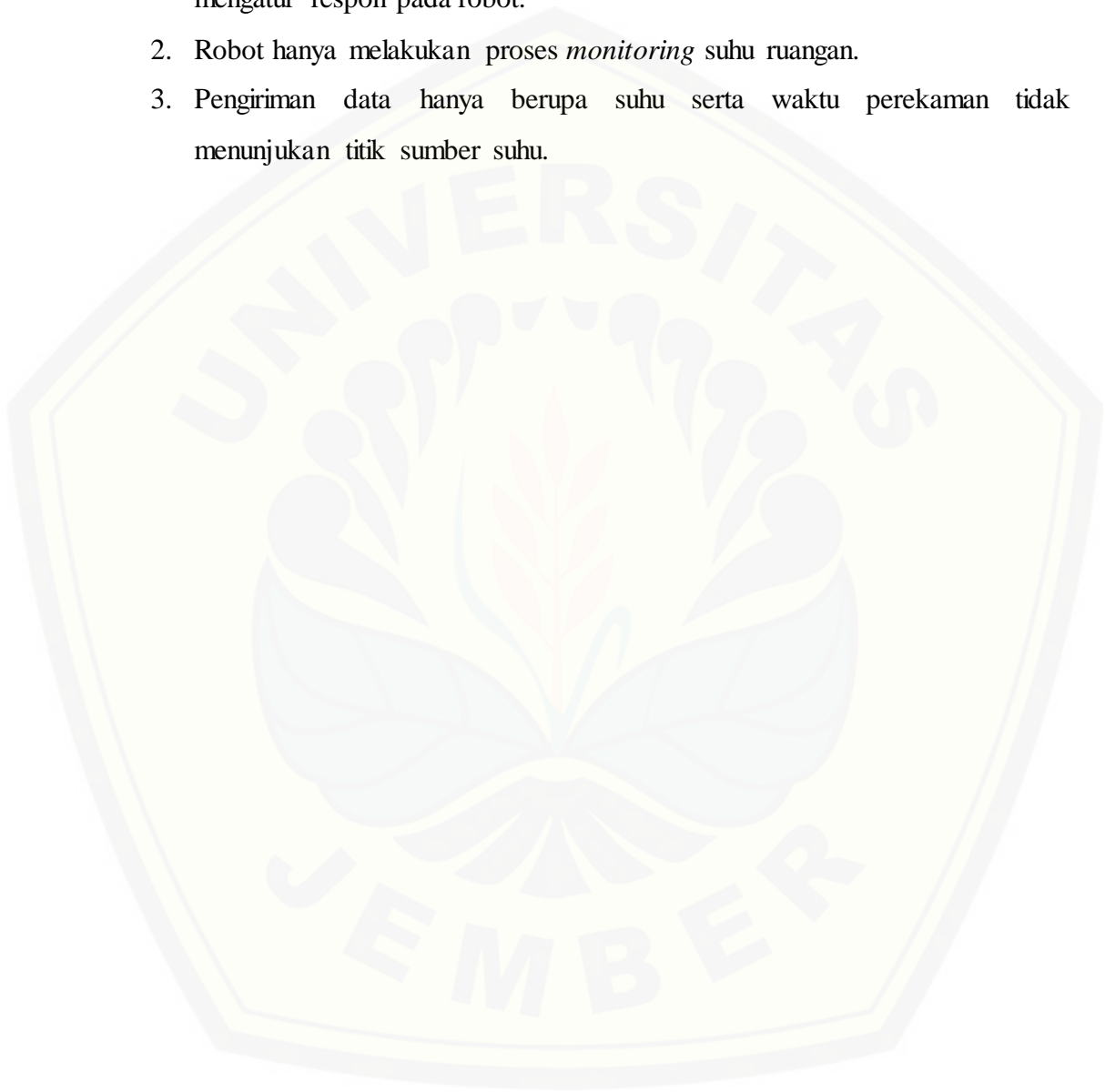
1. Dapat membuat sebuah robot otomatis untuk melakukan proses navigasi pada sebuah ruangan untuk *monitoring* suhu ruangan.
2. Dapat membuat kontrol kestabilan respon motor menggunakan kontrol *behavior based* dan PID.
3. Dapat melakukan pengujian robot diruangan sebenarnya.



#### 1.4 Batasan Masalah

Batasan Masalah dari penelitian ini antara lain:

1. Sitem kendali menggunakan kontrol *behavior based* dan PID untuk mengatur respon pada robot.
2. Robot hanya melakukan proses *monitoring* suhu ruangan.
3. Pengiriman data hanya berupa suhu serta waktu perekaman tidak menunjukkan titik sumber suhu.



## BAB 2. TINJAUAN PUSTAKA

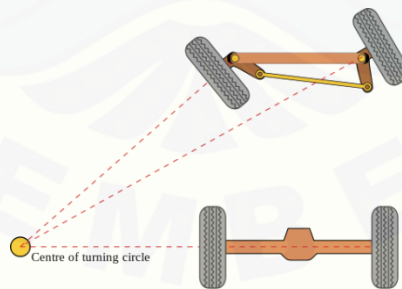
### 2.1 Robot Beroda

Robot adalah sebuah alat mekanik dan elektrik yang dapat melakukan tugas fisik, dengan atau tanpa pengawasan dari manusia. Robot biasa digunakan untuk membantu pekerjaan manusia dimana pekerjaan itu bersifat berat, berbahaya, pekerjaan yang memiliki kecepatan yang berulang dan kotor. (Muqorrobin, 2016)

Ada banyak jenis dari robot itu salah satunya yaitu robot beroda. Robot beroda atau (*Wheeled Mobile Robot*) adalah robot sebuah robot yang memiliki kemampuan untuk berpindah dari suatu titik ke titik yang lain menggunakan aktuator berupa roda. Roda itu sendiri dikendalikan untuk menentukan arah berjalannya robot. Adapun untuk jenis pengendaliannya antara lain:

#### 1. *Ackerman Steering*

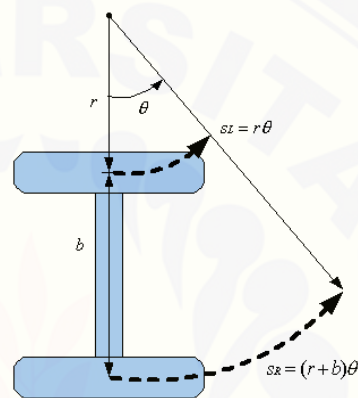
Pengendalian ackerman adalah bentuk pengendalian roda yang menggunakan empat roda depan dan belakang, untuk membelok kekanan atau kekiri digunakan roda deapan, dan dua roda dibelakang digunakan sebagai penompang dan pendongorong robot kedua roda ini diberi kecepatan yang sama.



Gambar 2.1 Pengendalian *ackreman steering*

## 2. *Differential steering*

Pengendalian *differential steering* adalah pengendalian pada robot yang memiliki dua buah roda. Untuk pengendalian *differential steering* menggunakan beda kecepatan antara roda kiri dan kanan. Robot dapat belok kiri apabila putaran roda kiri lebih pelan dibanding dengan roda kanan. Sedangkan untuk belok kanan putaran roda kanan lebih pelan dibanding roda kiri. (Roziyerus, 2013)



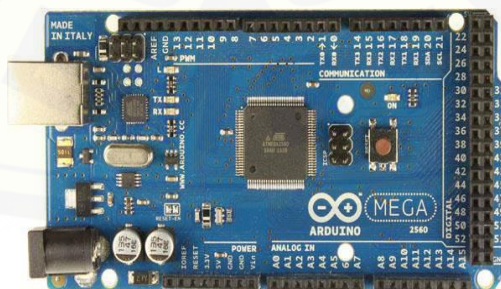
Gambar 2.2 Pengendalian *differential steering*

### 2.2 Arduino Mega 2560

Arduino Mega 2560 merupakan jenis – jenis dari arduino, sedangkan arduino adalah sebuah mikrokontroler yang memiliki sifat *open source* sehingga memudahkan dalam memakainya. Arduino memiliki prosesor Atmel AVR dan memiliki bahasa pemrograman sendiri serta banyak bantuan *library* pada arduino. Arduino mega 2560 memiliki chip ATmega 2560 serta banyak pin pada *board* arduino. Adapun spesifikasi dari arduino mega 2560 adalah : (Yuhardiansyah, 2016)

Tabel 2.1 Spesifikasi arduino mega 2560

Chip mikrokontroler	ATmega2560
Tegangan operasi	5V
Tegangan <i>input</i> (yang direkomendasikan, via jack DC)	7V - 12V
Tegangan <i>input</i> (limit, via jack DC)	6V - 20V
Digital I/O pin	54 buah, 6 diantaranya menyediakan PWM <i>output</i>
Analog <i>Input</i> pin	16 buah
Arus DC per pin I/O	20 mA
Arus DC pin 3.3V	50 mA
Memori <i>Flash</i>	256 KB, 8 KB telah digunakan untuk <i>bootloader</i>
SRAM	8 KB
EEPROM	4 KB
<i>Clock speed</i>	16 Mhz
Dimensi	101.5 mm x 53.4 mm
Berat	37 g



Gambar 2.3 Bentuk Fisik Arduino Mega 2560

( sumber :Arduino.com)

### 2.3 LCD

LCD (*Liquid Cristal Display*) merupakan salah satu jenis *display* elektronik. *Display* elektronik sendiri merupakan alat elektronika yang memiliki fungsi untuk menampilkan sumber informasi yang ingin ditampilkan baik secara karakter, grafik, maupun data. LCD (*Liquid Cristal Display*) dapat menampilkan suatu data memanfaatkan teknologi CMOS *logic* dengan memantulkan cahaya yang berada disekelilingnya.



Gambar 2.4 LCD

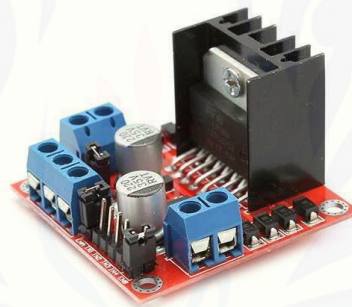
(Sumber : wulfden.org)

Di dalam modul LCD (*Liquid Cristal Display*) terdapat sebuah mikrokontroler yang memiliki fungsi untuk mengatur tampilan pada LCD (*Liquid Cristal Display*), didalam mikrokontroler ini dilengkapi dengan memori dan register. Adapun beberapa jenis memori itu adalah :

1. DDRAM (*Display Data Random Access Memory*) memori untuk tempak karakter yang akan ditampilkan.
2. CGRAM (*Character Generator Random Access Memory*) memori untuk menggambarkan pola karakter yang bentuknya dapat diubah.
3. CGROM (*Character Generator Read Only Memory*) memori untuk menggambarkan pola karakter yang sudah ada pada LCD (*Liquid Cristal Display*) dimana pola ini merupakan karakter bawaan dari LCD (*Liquid Cristal Display*) yang tidak dapat dirubah. (Elektronika Dasar, 2012a)

## 2.4 Driver Motor

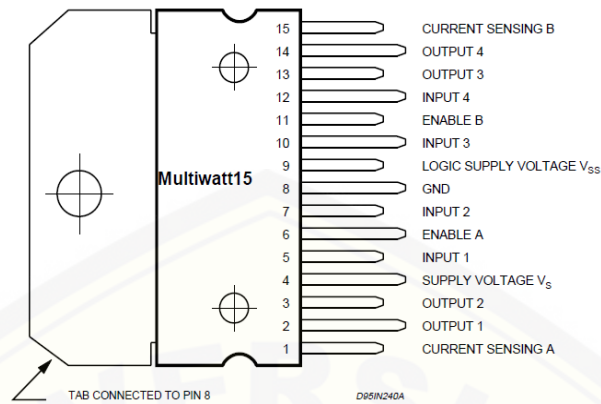
Motor DC merupakan komponen elektronika yang memerlukan sumber daya untuk menggerakkannya. Tidak semua motor dc dapat disambung langsung ke mikrokontroler ada yang memerlukan sumber daya dari luar untuk itu dibutuhkan *driver* motor yang dapat membantu mikrokontroler untuk mengendalikan motor dc. Selain itu *driver* motor menghindari arus balik dari motor ketika motor berhenti yang dapat membuat mikrokontroler itu sendiri rusak. Maka dari itu *driver* motor dapat membantu mencegah rusaknya mikrokontroler dari arus balik pada motor. Beberapa jenis *driver* motor yang sering digunakan diantaranya adalah *H-Bridge* transistor, H-Bridge MOSFET, dan IC *driver motor*.



Gambar 2.5 Modul *Driver Motor*

(Sumber : ectrnilab.com)

*Driver* motor juga dapat digunakan untuk mengontrol arah serta kecepatan putaran pada motor. Salah satu yang sering digunakan adalah modul IC *driver* motor L298 dalam modul IC ini terdapat rangkaian *H-Bridge* yang berguna untuk mengontrol putaran motor sesuai data masukan digital. Pada IC L298 terdapt pin untuk mengatur PWM (*Pulse Width Modulator*) yang akan mengatur kecepatan motor dc. Konstruksi pin *driver motor* DC IC L298 adalah sebagai berikut. (Suyadhi, 2015)



Gambar 2.6 Konfigurasi PIN *driver* motor L298

(Sumber :robotics-university.com)

## 2.5 Motor DC

Motor DC adalah motor listrik yang bergerak dengan suplai tegangan arus searah yang akan dirubah menjadi energi gerak mekanik. Bagian – bagian pada motor dc diantaranya adalah *stator* (bagian yang tidak berputar) dan disebut *rotor* (bagian yang berputar). Motro dc menggunakan arus langsung yang tidak langsung/*direct-unidirectional*. Bagian utama dari motor dc adalah :

1. Kutub medan, motor dc memiliki dua yaitu kutub utara dan kutub selatan.
2. Dinamo dihubungkan dengan as yang berfungsi untuk menggerakkan beban.
3. *Commutator* berfungsi untuk transmisi arus antara dinamo dan sumber daya.



Gambar 2.7 Motor DC

(Sumber: elektronika-dasar.web)

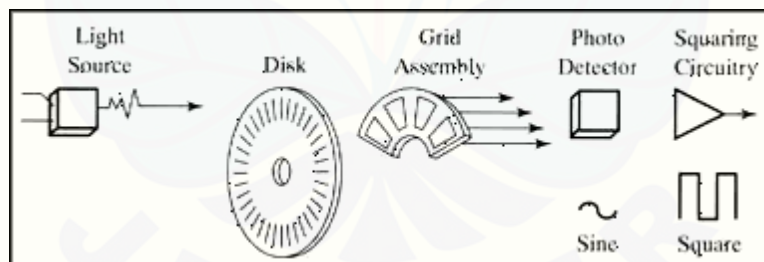
Beberapa keuntungan menggunakan motor dc adalah pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor dc dapat dikendalikan dengan mengatur :

1. Tegangan dinamo dengan meningkatnya tegangan maka kecepatan meningkat.
2. Arus medan dengan menurunkan arus maka kecepatan meningkat.

(Elektronika Dasar, 2012b)

## 2.6 Sensor Rotary Encoder

*Rotary encoder* adalah komponen elektronika yang berfungsi untuk *monitoring* gerak pada poros yang bergerak. *Rotary encoder* menggunakan sensor *optic* untuk membaca putaran untuk menghasilkan pulsa sehingga *rotary* mampu mengetahui jarak posisi dengan mengolah data putaran menjadi informasi berupa kode digital oleh *rotary encoder* yang akan diteruntukan ke kendali *encoder*. *Rotary encoder* merupakan sebuah piringan tipis yang memiliki lubang di lingkar piringannya. Piringan tipis itu sendiri disambung pada as roda motor sehingga diperoleh jumlah putaran pada motor.



Gambar 2.8 *Optical Rotary Encoder*

(Sumber : konversi.wordpress.com)

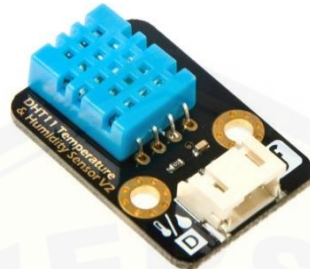
*Rotary encoder* memiliki beberapa jenis yang digunakan, yaitu *Absolute rotary encoder* dan *incremental rotary encoder*.

## 2.7 Sensor Suhu DHT 11

Sensor suhu DHT 11 adalah sebuah sensor digital yang berfungsi untuk mengetahui *temperature* dan kelembaban udara disekitar ruangan. Sensor DHT



merupakan sensor yang cukup baik karena respon dan stabilitasnya. Sensor ini memiliki *range temperature* 0 - 50°C. Serta *humidity* 20% - 90%. (Surono, 2013)



Gambar 2.9 Sensor Suhu DHT11

(Sumber : geraicerdas.com)

## 2.8 Esp 8266

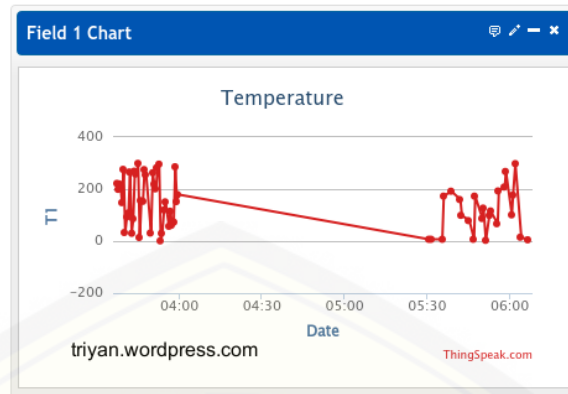
Esp 8266 adalah sebuah modul *Wi-Fi* yang dilengkapi dengan GPIO. Hal ini membuat ESP8266 dapat secara langsung menggantikan arduino dengan koneksi langsung dengan *Wi-Fi*. Modul esp8266 merupakan sebuah chip yang dapat menyambungkan dengan IoT (*internet of thing*). (ekoyulian, 2010)



Gambar 2.10 ESP 8266- 01

(Sumber : myelectronicslab.com)

IoT (*internet of thing*) diartikan dengan benda – benda yang dapat melakukan komunikasi satu sama lain menggunakan internet. Untuk menampilkan hasil komunikasi itu sendiri dibutuhkan *IoT Developer Program*. Salah satunya yaitu Thigspeak. ThingSpeak merupakan sebuah *platform Internet of Things* yang dapat digunakan atau diperoleh secara gratis untuk menampilkan *chart* suatu peralatan IoT dengan mengakses web dari [www. ThingSpeak.com](http://www.ThingSpeak.com). (triyani, 2015)



Gambar 2.11 Contoh grafik pada web thingspeak  
(Sumber : triyan.wordpress.com)

## 2.9 Buzzer

*Buzzer* adalah sebuah komponen elektronika yang memiliki untuk merubah energi listrik menjadi getaran suara. *Buzzer* sendiri biasanya digunakan sebagai sebuah indikator ketika terjadi kesalahan pada alat dan bisa jadi penanda ketika alat mendeteksi sesuatu. *Buzzer* memiliki frekuensi 1 – 5 KHZ dan bekerja pada tegangan 3 – 12 Volt. (Febri, 2015)

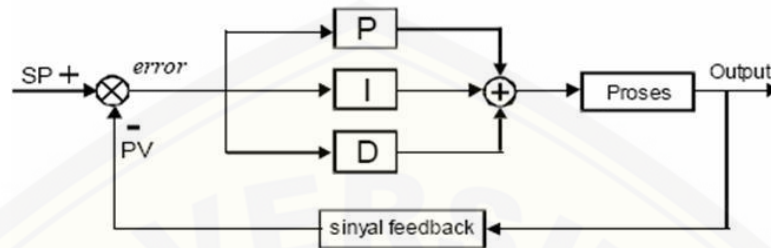


Gambar 2.12 Buzzer  
(Sumber : luckyretail.com)

## 2.10 Sistem Kendali PID (*Proportional Integral Derivative*)

Kontrol PID (*Proportional Integral Derivative*) adalah sebuah sistem kontrol yang digunakan untuk menambah tingkat kepresisian suatu sistem dengan menggunakan sistem umpan balik (*feedback*) yang terdapat dalam sistem tersebut.

Kontrol PID terbentuk dari tiga buah kontrol yaitu kontrol P (*Proportional*), D (*Derivative*) dan I (*Integral*). Dimana setiap kontrol memiliki karakteristik serta kelebihan dan kekurangan. Blok diagram sistem kendali PID ditunjukkan pada Gambar 2.13.



Gambar 2.13 Blok diagram pengendali PID

(Sumber : Bachri, 2004)

Adapun persamaan sistem kendali PID adalah :

$$PID = K_p \cdot e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p \cdot T_d \frac{de(t)}{dt} \quad (2.1)$$

Dengan :

PID = *output* dari kendali PID

$K_p$  = konstanta *proporsional*

$K_i = K_p/T_i$

$T_i$  = konstanta *integral*

$K_d = K_p \cdot T_d$

$T_d$  = konstanta *derivatif*

$e(t)$  = *error*

Untuk mendapatkan respon yang baik dari kontrol PID parameter harus mengatur masing-masing parameter P, I dan D.

### 2.10.1 Cara Mentuning Manual Parameter PID

Ada beberapa cara mentuning parameter dari PID salah satunya yaitu cara dengan coba-coba (*tranning and error*) dengan cara parameter-parameter PID diubah- ubah dengan melihat tabel respon PID untuk menentukan respon yang diinginkan. Dengan menganalisa respon PID, maka nilai-nilai  $K_p$ ,  $K_i$ , dan  $K_d$  bisa diubah-ubah sesuai dengan Tabel 2.2. Parameter PID ditunjukkan pada Tabel 2.2.

Tabel 2.2 Parameter PID

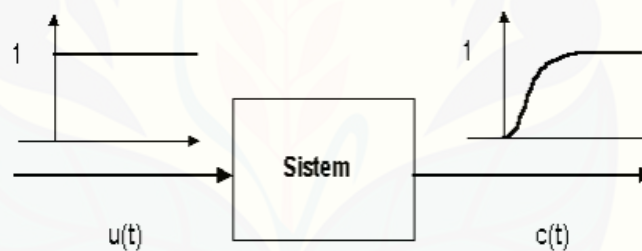
Parameter	Rise Timer	Overshoot	Setting Time	S-S Error
Kp	Berkurang	Bertambah	0	Berkurang
Ki	Berkurang	Bertambah	Bertambah	Menghilangkan
Kd	Minor Change	Berkurang	Berkurang	Minor Change

### 2.10.2 Metode Ziegler-Nichols

Kemudian ada juga dengan menggunakan metode *Ziegler-Nichols* metode ini adalah mencari parameter-parameter PID dengan rumus yang telah ditentukan. Metode ini memiliki 2 tipe, yaitu tipe 1 (*open loop*) dan tipe 2 (*closed loop*).

#### 2.10.2.1 Ziegler-Nichols tipe 1 (*open loop*)

*Ziegler-Nichol open loop*. Sistem diberi *input step* sehingga respon *open loop* terbentuk.

Gambar 2.14 Sistem Diberi *Input Step*

(Sumber : slideshare.net)

Kemudian dari respon *open loop*, diperoleh parameter-parameter ZN tipe 1 (L dan T). Proses desain menentukan parameter L dan T ditunjukkan pada Gambar 2.14.



Gambar 2.15 Proses Desain Penentuan Parameter L dan T

(Sumber : Bachri, 2004)

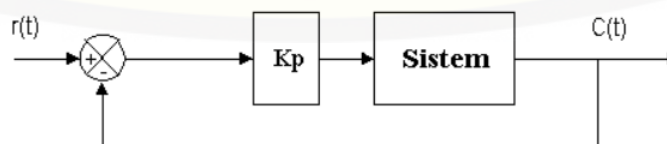
Kemudian ketika parameter L dan T diperoleh, nilai-nilai  $K_p$ ,  $T_i$ , dan  $T_d$  bisa dicari dengan menggunakan rumus PID untuk metode ZN tipe 1. Tabel 2.3 menunjukkan tabel parameter PID untuk ZN tipe 1.

Tabel 2.3 Parameter PID untuk ZN tipe 1

Type Of Controller	$K_p$	$T_i$	$T_d$
P	$T/L$	$\infty$	0
PI	$0.9T/L$	$L/0.3$	0
PID	$1.2T/L$	$2L$	0.51

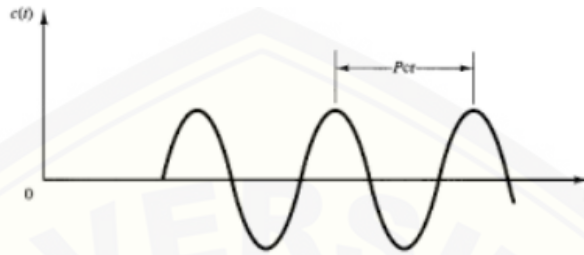
### 2.10.2.2 Ziegler-Nichols tipe 2 (closed loop)

Dalam metode ZN tipe 2, menggunakan sistem *closed loop* sebagai *feedback* respon. Pada metode ini yang digunakan  $K_p$  saja. Sistem dibuat hingga beresilasi terus menerus dengan mengatur besarnya nilai  $K_p$ .

Gambar 2.16 Sistem *closed loop* dengan Menggunakan  $K_p$ 

(Sumber : slideshare.net)

Besarnya nilai  $K_p$  saat respon sistem berosilasi terus menerus merupakan nilai  $K_{cr}$ . Dari respon yang dihasilkan, parameter lain ZN tipe 2 selain  $K_{cr}$ , yaitu  $P_{cr}$  dapat dicari. Proses desain menentukan parameter  $P_{cr}$  ditunjukkan pada gambar 2.14.



Gambar 2.17 Proses Mendesain Menentukan Parameter  $P_{cr}$

(Sumber : slideshare.net)

Setelah mendapatkan nilai parameter  $K_{cr}$  dan  $P_{cr}$ , nilai-nilai  $K_p$ ,  $T_i$ , dan  $T_d$  dapat dihitung dengan rumus mencari parameter PID untuk ZN tipe 2. Tabel 2.4 menunjukkan tabel parameter PID untuk ZN tipe 2. (Fauziansyah, 2015)

Tabel 2.4 Parameter PID Untuk ZN Tipe 2

Type Of Controller	$K_p$	$T_i$	$T_d$
P	$0.5K_{cr}$		0
PI	$0.45K_{cr}$	$(1/1.2)P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

## BAB 3. METODE PENELITIAN

### 3.1 Waktu dan Tempat

#### 3.1.1 Tempat

Adapun tempat penelitian, pengujian dan analisis dilakukan secara umum dilakukan di :

Penelitian skripsi dilakukan di salah satu lab dari jurusan Teknik Elektro Fakultas Teknik Universitas Jember yaitu Lab. Elektronika dan Terapan Fakultas Teknik Universitas Jember yang beralamatkan di Jl. Slamet Riyadi No. 62, Patrang, Kabupaten Jember. Lab Elektronika dan Terapan merupakan lab yang menyediakan alat-alat dan bahan berupa komponen elektronika serta sarana dan prasana untuk melakukan pengujian pada penelitian yang dilakukan yaitu penelitian mengenai sistem navigasi dari robot beroda untuk *monitoring* suhu ruangan sehingga membutuhkan suatu tempat pengujian dimana Lab Elektronika dan terapan merupakan tempat yang baik untuk melakukan pengujian dikarenakan dalam sebuah ruangan terdapat beberapa ruangan yang lain.

#### 3.1.2 Waktu

Adapun waktu penelitian, pengujian dan analisis dilakukan secara umum dilakukan Januari 2018 sampai dengan April 2018.

### 3.2 Alat dan Bahan

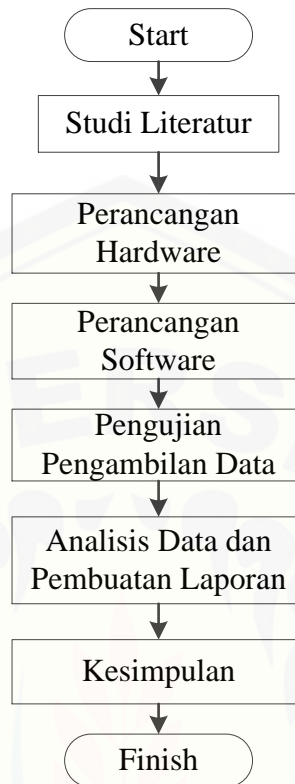
Dalam penelitian ini digunakan peralatan-peralatan sebagai penunjang pembuatan dan pengujian tugas akhir. Sedangkan untuk penjelasan beberapa komponen lebih lengkapnya sebagaimana dijelaskan dalam BAB 2. Alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut.

Tabel 3.1 Daftar Peralatan yang Digunakan

No	Nama Alat	Spesifikasi Teknis dan Penjelasan
1	Motor Dc Gearbox kuning	Tegangan <i>input</i> : dc 3-10 v Rasio reduksi: 1:48 Torsi: 0,8 kg cm
2	Arduino Controller	Arduino Mega2560. Atmega 2560 <i>microcontroller</i> , <i>Input voltage</i> 7-12V, 5V <i>Electric current</i> : 500mA, 3.3 <i>Electric current</i> : 50mA, 54 <i>Digital I/O Pins</i> (15 PWM <i>output</i> ), 16 <i>Analog Inputs</i> , 256 KB <i>Flash</i> <i>Memory</i> , 16 <i>Clock Speed</i>
3	Sensor Suhu	DHT11. <i>Input voltage</i> 3.5-5.5V, Rentang pengukuran kelembaban: 20% – 95% dengan toleransi +/- 5% Rentang pengukuran suhu: 0-50 °C, dengan toleransi +/- 2 °C.
4	LCD	LCD 2004 + I2C, <i>Voltage</i> 5V DC, <i>Number of</i> <i>Character</i> = 20 char x 4 line
5	Sensor Jarak HC-SR04	<i>Supply</i> tegangan 5V DC, Arus <i>Quiescent</i> < 2mA, Sudut efektif < 15 derajat, Jarak pengukuran 2 – 500 cm, Resolusi 0.3 cm.
6	Sensor <i>Rotary</i> <i>encoder</i>	Tegangan suplai DC 5V, <i>Output</i> digital, Logika <i>output</i> “1” : saat tidak mendeteksi lubang piringan Logika <i>output</i> “0” : saat mendeteksi lubang piringan.
7	<i>Driver</i> Motor L298	Tegangan suplai operasi sampai dengan 4-6V, Total arus DC yang mampu dilewatkan sampai dengan 4A Terdiri dari 2 bagian yang independen, Memiliki 4 <i>output</i> (dapat di koneksikan dengan 1 motor stepper, atau 2 motor DC 2 arah, atau 4 motor DC 1 arah), Dapat langsung dihubungkan pada mikrokontroller AVR, MCS-51, Arduino, ARM dsb.
8	<i>Buzzer</i>	<i>Input</i> DC 3-24V, Diameter: 4 cm, Suara: <i>Continue</i> / Bukan putus-putus
9	Esp 8266	<i>Vin</i> : DC 3,3V, 802.11 b / g / n, <i>Wi-Fi Direct</i> (P2P), <i>soft-AP</i> ), <i>Built-in TCP / IP protocol stack</i> , <i>Built-in TR</i> <i>switch</i> , <i>balun</i> , <i>LNA</i> , <i>power amplifier and matching</i> <i>network</i> , <i>Built-in PLL</i> , <i>voltage regulator and power</i> <i>management components</i> , 802.11b mode + 19.5dBm <i>output power</i> , <i>Built-in temperature sensor</i> , <i>Support</i> <i>antenna diversity</i> , <i>off leakage current is less than 10uA</i> <i>Built-in low-power 32-bit</i>



### 3.3 Metodologi Pelaksanaan dan Penelitian



Gambar 3.1 *Flowchart* Penelitian

Tahap penelitian “Rancang Bangun Sistem Navigasi Robot Beroda menggunakan Metode *Behavior Based* Untuk *Monitoring* Suhu Ruangan Dengan *Display* Android” adalah sebagai berikut :

1. Studi literatur

Tahap pertama melalui penelitian ini dilakukan dengan mencari literatur yang terkait dengan penelitian ini, dengan tujuan agar perancangan dan pelaksanaan penelitian dapat sesuai dengan yang diinginkan. Literatur diperoleh dari buku-buku dan jurnal yang berhubungan dengan sistem navigasi robot beroda, kontrol *Behavior Based* dan PID.

2. Perancangan *Hardware*

Tahap kedua adalah proses perancangan *hardware* baik dari alat dan bahan yang digunakan, desain robot beroda, rancangan bentuk dan cara kerja masing-masing komponen yang digunakan dan desain elektriknya.

### 3. Perancangan *Software*

Tahap ketiga adalah proses perancangan *plan* dan sistem kontrol PID serta kontrol *Behavior based*, program dan arduino serta pengiriman data menggunakan Esp8066 serta perancangan desain tampilan pada android.

### 4. Pengujian dan Pengambilan Data

Tahap keempat adalah pengujian terhadap komponen – komponen yang kritis seperti pengujian sensor suhu dan pengujian sensor jarak. Pengujian ini sangat penting karena dapat mempengaruhi tingkat keberhasilan dari proses navigasi pada robot.

### 5. Analisa Data dan Pembuatan Laporan

Tahap ke lima adalah melakukan analisa pada data yang diperoleh dari data sebelumnya. Pengambilan data dilakukan dari hasil proses navigasi robot dengan kontrol PID. Sehingga di harapkan menghasilkan keluaran yang diinginkan dengan menganalisa masing-masing percobaan.

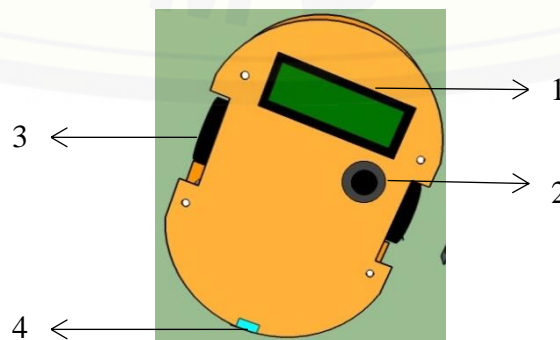
### 6. Penarikan Kesimpulan

Tahap ke enam adalah tahap terakhir pada penelitian yaitu memberikan kesimpulan dari pengujian dan analisa data yang diperoleh.

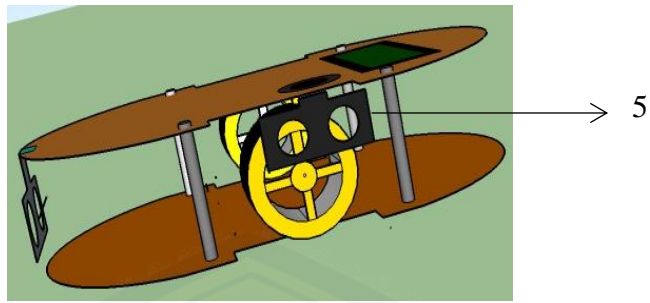
## 3.4 Perancangan Perangkat Keras

### 3.4.1 Desain Robot Beroda

Desain pembuatan mekanik robot beroda pemantau suhu ruangan ini yaitu terdiri dari *software* program arduino, pembuatan badan robot beroda, desainnya seperti dibawah ini:



Gambar 3.2 Robot tampak atas



Gambar 3.3 Robot tampak kiri atas



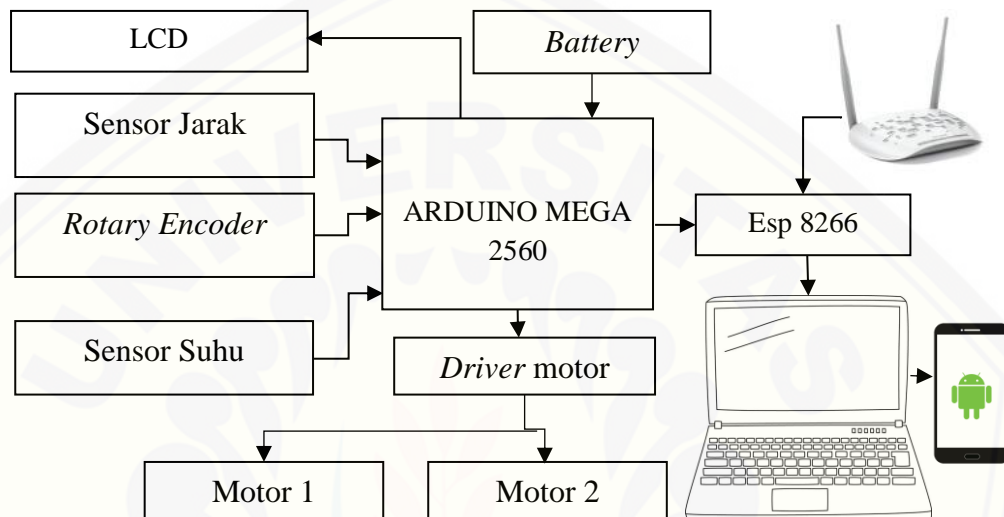
Gambar 3.4 Robot tampak depan

Keterangan:

1. LCD 20 x 4
2. *Buzzer*
3. Motor DC
4. DHT 11
5. Sensor Jarak

### 3.4.2 Desain Elektronik

Adapun desain elektronika robot beroda untuk monitoring suhu ruangan dengan *display* android yang memiliki *input*, *process* dan *output* seperti dibawah ini :



Gambar 3.5 Desain Elektronik

Gambar 3.5 adalah desain elektronik tentang blok diagram diagram robot beroda pemantau suhu dimana setiap blok disusun menjadi satu sistem pada robot pemantau suhu dengan metode *behavior based control*. Dalam proses pembuatan robot beroda pemantau suhu dengan metode *behavior based control* memiliki 3 bagian yaitu *Input*, *Process* dan *Output*. Kemudian penjelasan masing – masing bagian :

1. *Input*, Bagian *Input* terdiri dari sensor terdapat sensor jarak berguna sebagai informasi robot untuk keadaan disekitarnya. Kemudian ada sensor *rotary encoder* sensor ini berguna untuk menjaga kestabilan kecepatan motor. Ada sensor suhu disini sensor suhu digunakan untuk mendeteksi suhu sekitar robot untuk proses monitoring itu sendiri.

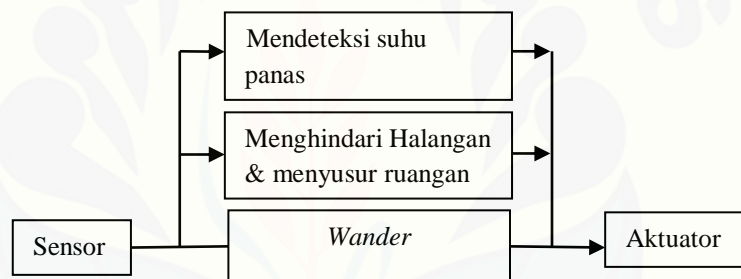
2. *Process*, Bagian *Process* terdapat arduino, arduino disini sebagai pengendali untuk memproses informasi *input* menuju *output*. *Process* disini menggunakan metode *Behavior Based Control*.

3. *Output*, Bagian *Output* terdapat 2 motor DC, Esp 8266, dan LCD. Motor DC digunakan sebagai penggerak robot. Esp 8266 digunakan sebagai pengirim data ke android. Kemudian LCD digunakan untuk menampilkan suhu sekitar robot.

### 3.5 Sistem Kendali Robot *Behavior Based Control*

#### 3.5.1 *Behavior Based Control*

Pada perancangan sistem kontrol *behavior based control*, masing – masing keputusan dilakukan sesuai dengan tingkatan *behavior* yang telah diatur sesuai dengan keutamaannya. Dimana digunakan sebuah koordinator yang digunakan untuk mengatur setiap perilaku pada setiap satu *behavior* tersebut.



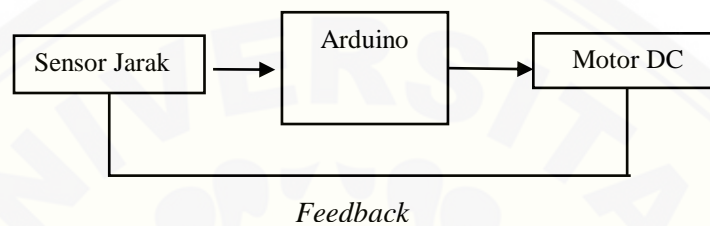
Gambar 3.6 Diagram blok *Behavior based control*

Dari blok diagram dijelaskan bahwa fungsi dari koordinator yaitu sebagai pengambil keputusan untuk *behavior* pada robot mengacu pada lingkungan serta kebutuhan robot. Pada *input* yang diterima dari sensor menentukan *behavior* yang dilaksanakan robot. Perilaku pertama pada robot yaitu melakukan *eksplorasi* secara bebas dimana robot akan bergerak secara acak atau linglung. Ketika ada mendeteksi halangan atau tembok maka robot akan melakukan perilaku yang kedua yaitu menghindari halangan dan ketika ada tembok robot akan menyusuri ruangan dengan cara menyusur pada tembok. Kemudian ketika robot saat melakukan proses *monitoring* mendeteksi suhu panas yang tinggi maka robot akan membunyikan *buzzer* selama 5 detik dan akan berhenti sesaat untuk mengirimkan data.

### 3.5.2 Bagian *Behavior-Based Control*

#### 3.5.2.1 *Wander*

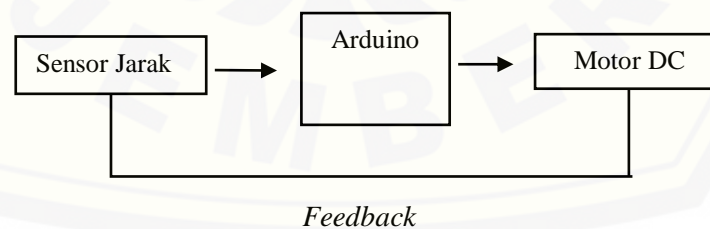
Prilaku *wander* ruangan ini akan terjadi ketika robot dalam keadaan jauh dari tembok dimana robot itu sendiri akan bergerak acak. Pada saat sensor jarak membaca jarak yang jauh maka informasi dari jarak dan akan diproses oleh arduino kemudian arduino akan mengatur kecepatan pada motor DC.



Gambar 3.7 Diagram blok *wander*

#### 3.5.2.2 Menghindari halangan dan menyusur ruangan

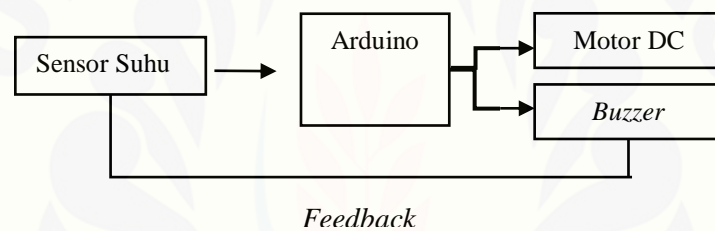
Perilaku menghindari halangan dan menyusur ruangan ini akan terjadi apabila ketika robot mendeteksi adanya halangan pada jarak tertentu dimana informasi tersebut akan diinformasikan ke arduino sebagai mikrokontroler untuk proses pada kecepatan motor DC. *Behavior* menyusur ruangan ini akan membuat robot mampu menyusuri ruangan dengan cara menyusur pada tembok kemudian ketika disekitar robot ada halangan robot mampu untuk menghindarinya.



Gambar 3.8 Diagram blok menghindari halangan dan menyusur ruangan

### 3.5.2.3 Mendeteksi suhu panas

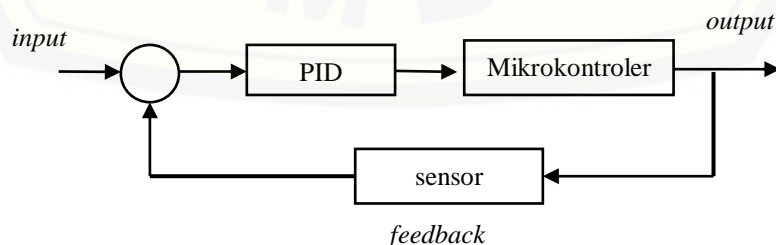
Perilaku menemukan titik panas ini akan terjadi apa bila saat robot beroda menyusur ruangan dan melakukan proses *monitoring* suhu ruangan dan saat robot mendeteksi suhu ruangan yang tinggi maka informasi akan disampaikan ke arduino sebagai mikrokontroler dan robot akan secara otomatis membunyikan *buzzer*, robot akan tetap berjalan untuk melakukan proses *monitoring*. Untuk proses perekaman suhu sendiri dilakukan secara terus menerus sedangkan untuk proses pengiriman data suhu ke web thingspeak.com dilakukan setiap 30 detik sekali dalam proses pengiriman itu sendiri robot akan berhenti sejenak dan setelah dilakukan pengiriman robot akan berjalan kembali.



Gambar 3.9 Diagram blok menemukan suhu panas

### 3.6 Sistem Kontrol Robot menggunakan PID

Pada rancang bangun robot beroda menggunakan metode *behavior based* untuk proses *monitoring* suhu ruangan dengan *display* android sendiri menggunakan kontrol PID sebagai kontrol dari aktuator pada robot dalam hal ini yaitu motor Dc sebagai penggerak dari robot. Kontrol PID sendiri berfungsi sebagai penstabil kecepatan motor untuk proses dari navigasi robot.



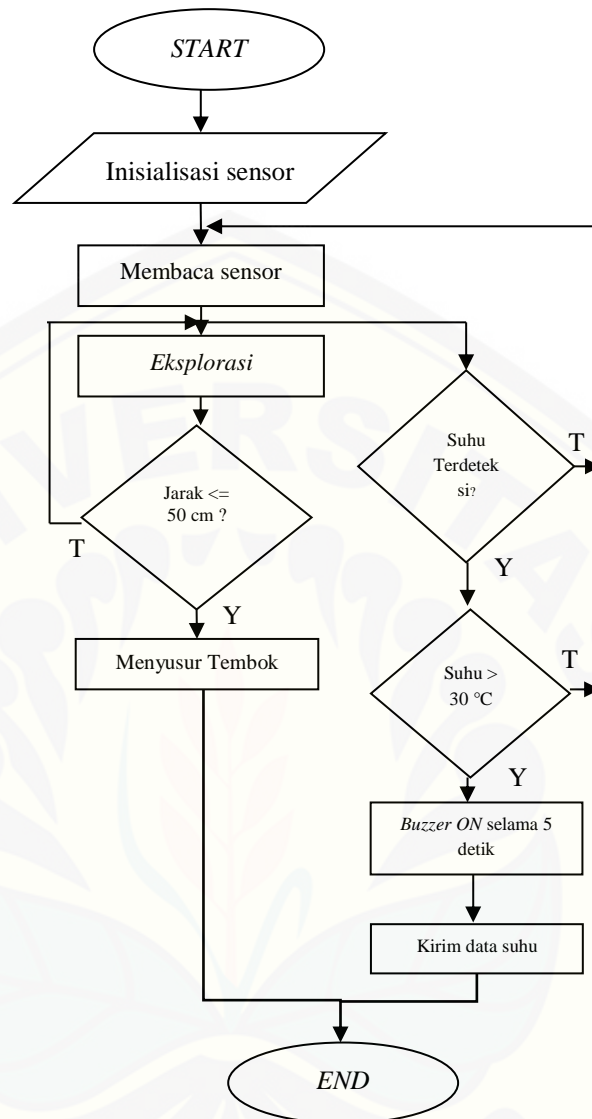
Gambar 3.10 Diagram blok kontrol PID

Pada gambar 3.10 diagram blok kontrol PID dapat dijelaskan bahwa kontrol PID di set didalam miktokontroler dimana *output* dari mikrokontroler yang berupa kecepatan pada motor DC itu sendiri digunakan sebagai *feedback* menggunakan sensor untuk memenuhi *input* berupa *set point* atau target yang telah ditentukan.

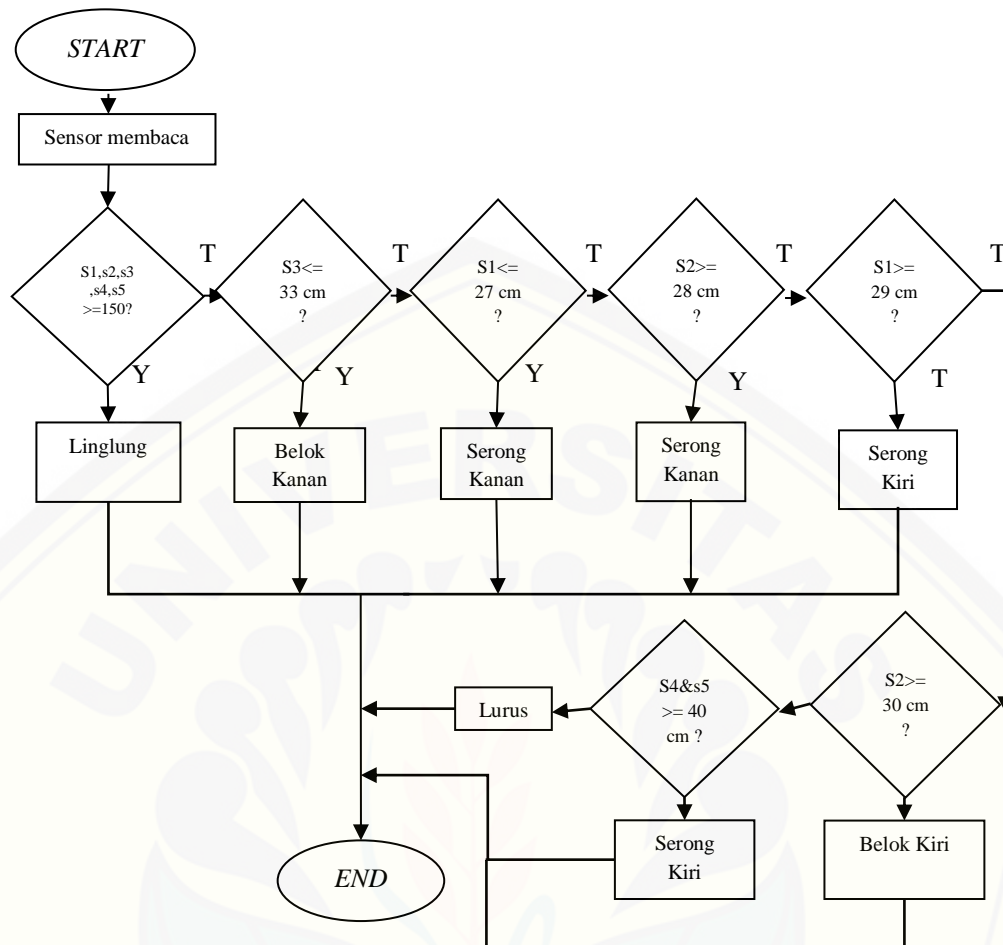
### 3.7 Flowchart

Pada *flowchart* gambar 3.11 menjelaskan tentang kerja robot beroda pemantau suhu ruangan menggunakan metode *Behavior Based Control*. Dimana ketika robot dalam kondisi awal robot akan mendapatkan informasi dari sensor jarak, suhu, dan *rotary encoder*. Ketika jarak halangan jauh maka robot akan melakukan proses berkeliling secara acak disini robot akan linglung. Kemudian ketika jarak halangan atau tembok dekat maka robot akan menyusuri tembok dan menyusur ruangan. Ketika menyusur ruangan robot mendeteksi suhu panas maka robot akan membunyikan *buzzer* selama 5 detik dan robot akan berhenti setiap 30 detik sekali untuk mengirimkan data suhu yang telah dibaca oleh sensor. Kemudian robot akan melanjutkan proses *monitoring* ruangan. Selama proses itu robot akan terus melakukan proses *monitoring* suhu ruangan kemudian data suhu itu akan dikirim melalui modul wifi esp 8266 ke android.





Gambar 3.11 Flowchart keseluruhan



Gambar 3.12 Flowchart menyusur tembok

### 3.8 Metode Perancangan Pengujian

Pengujian robot beroda pemantau suhu ruangan dibagi menjadi dua yaitu pengujian metode *behavior based control*, dan pengujian keseluruhan.

#### 3.8.1 Perancangan Pengujian Motor Dc

Pada pengujian tahap ini yaitu pengujian terhadap aktuator dimana kita dapat mengetahui pergerakan motor yang digunakan sebagai penggerak dari robot.

#### 3.8.2 Perancangan Pengujian Eps 8266

Pada pengujian esp 8266 bertujuan untuk menguji koneksi pengiriman pada esp 8266. Pengujian ini dilakukan agar mengetahui kecepatan pengiriman data serta banyaknya data yang mampu dikirim pada setiap waktu.

Pada pengujian esp 8266 dilakukan dua jenis pengujian pada esp 8266 yaitu saat LOS (*Line Off Sight*) yaitu pengujian kondisi ketika pengirim (Tx) dan penerima (Rx) berhadapan secara langsung, berarti esp 8266 dan *wi-fi server*

langsung berhadapan tanpa ada halangan. Pada pengujian ini diperoleh hubungan antara jarak antara *wi-fi server* dengan esp 8266 terhadap banyaknya data yang terkirim serta kecepatan terkirimnya data. Kemudian pengujian saat NLOS (*Near Line Off Sight*) yaitu ketika pengirim (Tx) dan penerima (Rx) saat berkomunikasi terhalang oleh sesuatu pada pengujian ini terhalang oleh tembok. Jadi saat melakukan pengujian esp 8266 diletakkan dalam ruangan tertutup sedangkan *wi-fi server* berada diluar ruangan. Ketika pengujian NLOS ini akan diketahui pengaruh dari halangan untuk proses pengiriman banyaknya data serta kecepatan pengiriman data.

### **3.8.3 Perancangan Pengujian Kontrol PID**

Pada pengujian kontrol PID diharapkan motor mampu bergerak sesuai dengan respon yang diharapkan. Untuk itu dibutuhkan proses *tunning* pada parameter PID. Pada pengujian PID dilakukan beberapa pengujian yaitu pengujian PID tanpa beban, pengujian PID dengan beban, serta pengujian PID pada rantai berpasir untuk mengetahui respon dari motor untuk keperluan gerak pada robot beroda.

### **3.8.4 Perancangan Pengujian *Behavior Based Control***

Pengujian metode *behavior based control* membahas tentang tiap perilaku dari metode tersebut. Pengujian berawal dari proses *eksplorasi* atau berkeliling, selanjutnya pengujian menghindari halangan serta menyusur ruangan, setelah itu perilaku ketika robot menemukan suhu panas. Tahap ini fokus tiap bagian-bagian dari metode *behavior based control*.

### **3.8.5 Perancangan Pengujian Keseluruhan**

Pengujian keseluruhan meliputi pengujian bagian dan *behavior based control*. Pada pengujian ini meliputi cara kerja robot secara keseluruhan. Dimana pengujian ini mengacu pada parameter metode *behavior based control* yang telah dibuat. Pada pengujian ini semua *behavior* yang telah diprogram disatukan untuk melihat pergerakan robot secara keseluruhan serta melihat kinerja robot secara keseluruhan.

### 3.9 Hasil Perancangan Alat

Hasil perancangan alat adalah gambar dari alat keseluruhan yang telah dibuat secara nyata sesuai dengan desain perancangan alat pada gambar 3.2, 3.3, dan 3.4. Berikut hasil perancangan alat secara keseluruhan :



Gambar 3.13 Robot tampak atas



Gambar 3.14 Robot tampak samping



Gambar 3.15 Robot tampak depan

### 3.10 Hasil Perancangan Aplikasi Pada Android

Hasil perancangan aplikasi pada android menggunakan aplikasi Virtuino yang sudah tersedia di *google play store*, dimana aplikasi ini akan menampilkan suhu yang terbaca oleh robot yang telah dikirim ke web *thingspeak.com* berupa angka dan grafik. Langkah pembuatan tampilan aplikasi yaitu :

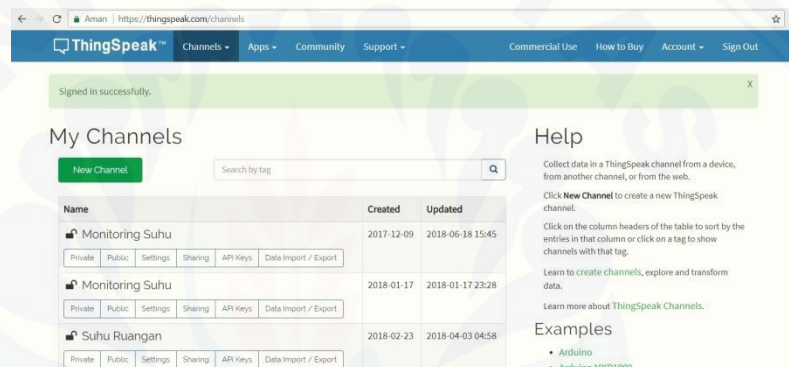
Proses pembuatan tampilan pada android yang pertama yaitu pembuatan akun di web *thingspeak.com* sebagai *platform Internet of Things* yang dapat digunakan secara gratis untuk menampilkan *chart* suatu peralatan IoT.

a. Membuka web thingspeak.com dan membuat akun.



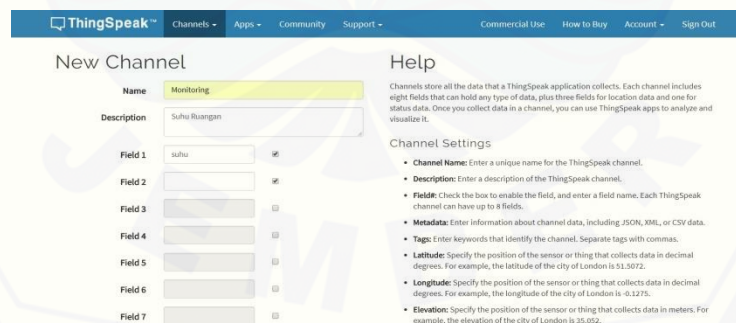
Gambar 3.16 Tampilan awal web

b. Membuat *channel* untuk media penyimpanan data.



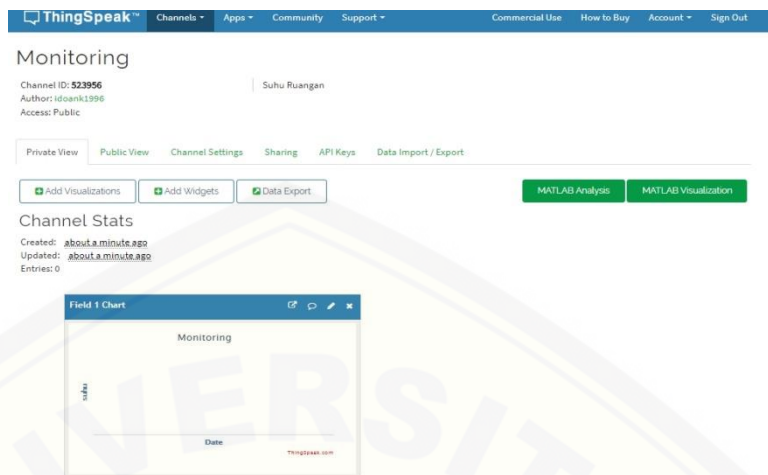
Gambar 3.17 Tampilan awal setelah masuk pada web

c. Pengaturan saat pembuatan *channel* dan simpan *channel* yang dibuat.



Gambar 3.18 Tampilan pada *channel*

d. Tampilan akhir pada *channel* yang dibuat



Gambar 3.19 Tampilan akhir *channels*

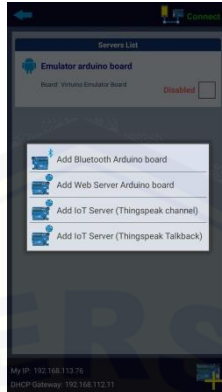
Tahap selanjutnya pembuatan tampilan pada android dengan menggunakan aplikasi virtuino yang telah di *download* dan terpasang pada perangkat android.

a. Tampilan awal aplikasi virtuino



Gambar 3.20 Tampilan awal virtuino

b. Pengaturan sambungan aplikasi dengan web thingspeak.com pada pojok kiri kemudian klik gambar pojok kanan bawah dan pilih *add IoT server*.

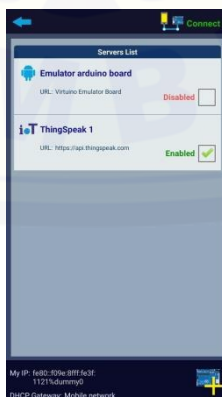


Gambar 3.21 Tampilan awal pengaturan sambungan

c. Setelah memilih *add IoT server* akan muncul tampilan untuk pengaturan sambungan dengan memasukkan cara memasukkan *channel id* yang telah dibuat pada web thingspeak.com, kemudian *save* pada pojok kanan atas.

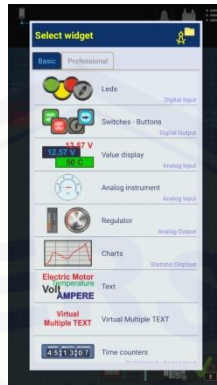


Gambar 3.22 Tampilan saat pengaturan sambungan



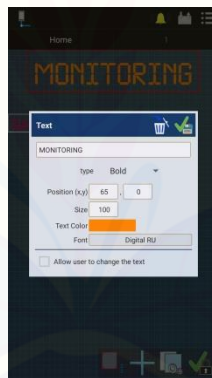
Gambar 3.23 Tampilan ketika aplikasi sudah tersambung

d. Pembuatan *text* untuk judul serta *text* untuk informasi tampilan android dengan cara memilih *select widget* berupa tanda tambah di pojok kiri lalu muncul beberapa pilihan dan pilih *text*.



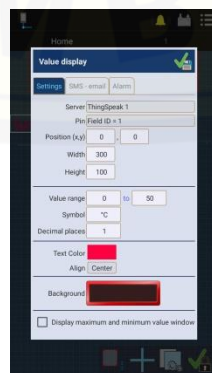
Gambar 3.24 Menu awal pembuatan tampilan

e. Setelah itu muncul tampilan di menu *text*, isi informasi yang dibutuhkan dan pilih *save*.



Gambar 3.25 Tampilan pengaturan untuk tampilan *text*

f. Pembuatan tampilan angka untuk data suhu, dengan cara memilih *value display* pada menu *select widget*, kemudian isi informasi yang dibutuhkan.



Gambar 3.26 Tampilan pengaturan untuk tampilan angka



g. Pembuatan tampilan berupa grafik untuk data suhu, dengan cara memilih *chart* pada menu *select widget*, kemudian isi informasi yang dibutuhkan.



Gambar 3.27 Tampilan pengaturan untuk tampilan grafik

h. Untuk tahap akhir pembuatan aplikasi untuk tampilan pada android *lock project* dengan cara klik pojok kanan bawah berupa gambar gembok agar tampilan tidak dapat berubah dan dapat juga ditambah gambar sebagai hiasan pada tampilan.



Gambar 3.28 Hasil akhir tampilan android

## BAB 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dari hasil yang telah diperoleh dalam penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Robot beroda dengan sistem kontrol *behavior based* untuk *monitoring* suhu ruangan ini cukup bagus dalam mengerjakan tugasnya. Robot dapat melakukan pengukuran suhu serta penyimpanan data suhu kedalam sebuah web dan mampu ditampilkan pada sebuah aplikasi pada android dimana robot memiliki presentase keberhasilan mencapai 60% saat melakukan proses perekaman data suhu, serta mampu mendeteksi perubahan suhu ruangan.
2. Sistem kontrol PID untuk membantu proses navigasi didalam ruangan cukup bagus digunakan untuk meningkatkan respon pada motor DC sebagai aktuator robot, respon motor sendiri memiliki  $t_r$  7,3 s,  $t_s$  7,3 s, dan  $M_p$  2 %.
3. Sistem kontrol *behavior based* untuk pengendali robot *monitoring* suhu ruangan, robot mampu menyelesaikan proses navigasi secara baik dengan presentase keberhasilan 60% dan meningkat menjadi 70% ketika dikhususkan pada titik navigasi dari ruangan 1 (satu) menuju ruang 2 (dua).

### 5.2 Saran

Dengan harapan penelitian ini dapat dikembangkan dan disempurnakan dikemudian hari, penulis memberikan saran sebagai berikut:

1. Robot beroda pemantau suhu ruangan ini masih perlu perbaikan karena masih terjadi beberapa kali tabrakan serta kebingungan pada robot, saat proses navigasi. Disarankan menggunakan sensor yang lebih bagus semisal kamera dll, untuk membantu robot dalam proses navigasi.
2. Sistem kontrol PID yang digunakan belum cukup bagus untuk respon robot secara langsung dipenerapkan saat proses navigasi robot dikarenakan spesifikasi motor yang kurang sehingga disarankan menggunakan spesifikasi motor yang lebih bagus dari segi rpm maupun torsi.

**DAFTAR PUSTAKA**

- Ali, M. 2004. Pembelajaran perancangan sistem kontrol pid dengan software matlab. *Jurnal Edukasi@Elektro*. 1(1):1–8.
- Bedogni, L. 2016. Arduino and esp8266
- D-robotics. 2010. Dht 11 humidity & temperature sensor. 9.
- Gunarta, L. 2011. Sensor ultrasonik. *Journal of Chemical Information and Modeling*. 53(9):1689–1699.
- Kariyanto, J. D., A. H. Alasiry, F. Ardila, dan N. Hanafi. 2011. Navigasi mobile robot berbasis trajektori dan odometry dengan pemulihan jalur secara otomatis. 1–8.
- Nurmaini Siti, Z. A. 2009. Sistem navigasi non holonomic mobile robot menggunakan aplikasi sensor ultrasonic. *Jurnal Ilmiah Generic*. 4(1):11.
- Nurshauma Syuhri, I. 2017. Robot soccer beroda dengan metode behavior based control. 160.
- Rijal, M. khoirul. 2016. Digital Repository Universitas Jember Digital Repository Universitas Jember
- Rohman, fathor. 2017. Rancang bangun alat pengering tembakau otomatis dengan metode pengolahan citra digital menggunakan kontrol hybrid fuzzy-pid. 83.
- Support, P. dan C. Service. 2014. Tutorial - l298n dual motor controller module 2a and arduino
- Syam, R. dan W. Erlangga. 2012. Rancang bangun omni wheels robot dengan roda penggerak independent. 3(1):213–220.
- Taufik, A. S. 2013. Sistem navigasi waypoint pada autonomous mobile robot. *Jurnal Mahasiswa TEUB*. 1(1):1–6.



# LAMPIRAN

**A. Listing Program Arduino**

```

#include <TimerOne.h>
#include <stdlib.h>
#define SSID "NIL" //nama wifinya
#define PASS "NenyIndah" //password wifi routernya
#define Baud_Rate 115200
#define Delay_Time 5000
#define WRITE_APIKEY "SJGY2OVQV2G3LD5T"//channel read apikey
#include <DHT11.h>
DHT11 dht11(26);
int err;
float temp, humi;
bool kirimdata; //cek status kirim data
double lpdKiri,lpdKanan,
    nilaiLPDKiri,nilaiLPDKanan,
    errorKiri, errorKanan,
    targetKiri, targetKanan,
    pwm, pwm2,
    i,
    kp=0.025, ki=0.001, kd=0.1, lasterrorKiri,I,
    kp2=0.025, ki2=0.001, kd2=0.1, lasterrorKanan,I2;

#define motorKa
#define motorKi
#define I1 8
#define I2 9
#define r1 10
#define r2 11
int trigkil = 27;
int echokil = 25;
int trigki2 = 31;
int echoki2 = 29;
int trigdepan = 35;
int echodepan = 33;
int trigka2 = 39;
int echoka2 = 37;
int trigka1 = 49;
int echoka1 = 47;
long s1, s2, s3, s4, s5 ;
long waktu;
int j;
void setup() {
    pinMode(27,OUTPUT);
    pinMode(31,OUTPUT);
    pinMode(35,OUTPUT);
    pinMode(39,OUTPUT);
    pinMode(49,OUTPUT);

```

```
pinMode(A0,OUTPUT);

pinMode(25,INPUT_PULLUP);
pinMode(29,INPUT_PULLUP);
pinMode(33,INPUT_PULLUP);
pinMode(37,INPUT_PULLUP);
pinMode(47,INPUT_PULLUP);

Serial.begin(9600);
pinMode(12,OUTPUT);
pinMode(11,OUTPUT);
pinMode(10, OUTPUT);
pinMode(9,OUTPUT);
pinMode(8, OUTPUT);
Serial3.begin(Baud_Rate);
Serial3.println("AT");
delay(5000);
//sambungkan ke wifi
Serial3.println("AT+CWMODE=1");//mode wifi client
delay(2000);
String cmd = "AT+CWJAP=\"";//at command konek ke wifi router
cmd += SSID;
cmd += "\",\"";
cmd += PASS;
cmd += "\"";//simpelnya ya gini AT+CWJAP="SSID","PASS";

Serial3.println(cmd);
delay(5000);

attachInterrupt(digitalPinToInterrupt(3),tambahKiri,RISING);
attachInterrupt(digitalPinToInterrupt(2),tambahKanan,RISING);
Timer1.initialize(1000000); // set a timer of length 100000 microseconds (or 0.1 sec - or
10Hz => the led will blink 5 times, 5 cycles of on-and-off, per second)
Timer1.attachInterrupt(waktuHabis);
delay (150);
}
void loop() {
if (waktu + 30000 < millis())
{
//digitalWrite(A0,HIGH);
motor (0,0);
 kirim();
Serial.println("KIRIM");
//delay(1000);
waktu = millis ();
//digitalWrite(A0,LOW);
}
}
dhtProg();
```

```
kiri1();
kiri2();
depan();
kanan2();
kanan1();
Serial.println(temp);
Serial.println (String(s1)+" "+String(s2)+" "+String(s3)+" "+String(s4)+" "+String(s5));
if (s1>= 150 && s2>= 150 && s3>= 150 && s4>= 150 && s5>= 150){
    linglung();
    Serial.println ("linglung");}
else if(s3<= 33){
    belokkanan();
    Serial.println ("belokkanan");}
else if (s1<= 27){
    serongkanan();
    Serial.println ("serongkanan");
}
else if (s2<= 28){
    serongkanan();
    Serial.println ("serongkanan");}
else if (s1>= 29){
    serongkiri();
    Serial.println ("serongkiri");}
else if (s2>= 30){
    belokkiri();
    Serial.println ("belokkiri");}
else if (s4 < 40 || s5 < 40){
    serongkiri();
}
else {
    lurus();
    Serial.println ("lurus");
}
}

void linglung(){
    j++;
    Serial.println(j);
    if(j==65)j=0;
    if(j<=13){
        jalan(75,150);
        Serial.println(" KIRI");
    }
    else if(j<25){
        jalan(155,150);
        Serial.println(" LURUS");
    }
    else if(j<38){
```

```
    jalan(130,50);
    Serial.println(" KANAN");
  }
  else if(j<60){
    jalan(155,150);
    Serial.println(" LURUS");
  }
  //delay(10);
}

void kiri1(){
  long durationki1;
  digitalWrite(trigki1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigki1, HIGH);
  delayMicroseconds(5);
  digitalWrite(trigki1, LOW);
  durationki1 = pulseIn(echoki1, HIGH,5000);
  s1 = durationki1/29/2;
  if (s1==0)s1=300;
  delay (10);
}
void kiri2() {
  long durationki2;
  digitalWrite(trigki2, LOW);
  delayMicroseconds(2);
  digitalWrite(trigki2, HIGH);
  delayMicroseconds(5);
  digitalWrite(trigki2, LOW);
  durationki2 = pulseIn(echoki2, HIGH,5000);
  s2 = durationki2/29/2;
  if (s2==0)s2=300;
  delay (10);
}
void depan(){
  long durationdepan;
  digitalWrite(trigdepan, LOW);
  delayMicroseconds(2);
  digitalWrite(trigdepan, HIGH);
  delayMicroseconds(5);
  digitalWrite(trigdepan, LOW);
  durationdepan = pulseIn(echodepan, HIGH,5000);
  s3 = durationdepan/29/2;
  if (s3==0)s3=300;
  delay (10);
}
void kanan2(){
  long durationka2;
```



```
digitalWrite(trigka2, LOW);
delayMicroseconds(2);
digitalWrite(trigka2, HIGH);
delayMicroseconds(5);
digitalWrite(trigka2, LOW);
durationka2 = pulseIn(echoka2, HIGH,5000);
s4 = durationka2/29/2;
if (s4==0)s4=300;
delay (10);
}
void kanan1(){
  long durationka1;
  digitalWrite(trigka1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigka1, HIGH);
  delayMicroseconds(5);
  digitalWrite(trigka1, LOW);
  durationka1 = pulseIn(echoka1, HIGH,5000);
  s5 = durationka1/29/2;
  if (s5==0)s5=300;
  delay (10);
}
void belokkiri(){
  jalan (50,150);
}
void serongkiri(){
  jalan (70,120);
}
void lurus(){
  jalan (155,150);
}
void serongkanan(){
  jalan (120,75);
}
void belokkanan(){
  jalan (150,55);
}

void motor(int kecKiri, int kecKanan){
  if (kecKiri>255) kecKiri=255;
  else if (kecKiri<-255) kecKiri=-255;
  if (kecKanan>255) kecKanan=255;
  else if (kecKanan<-255) kecKanan=-255;

  if(kecKanan>=0){
    digitalWrite(11,LOW);
    analogWrite (12,kecKanan);
  }
}
```

```
else{
    digitalWrite(l2,LOW);
    analogWrite (l1,0-kecKanan);
}
if(kecKiri>=0){
    digitalWrite(r2,LOW);
    analogWrite (r1,kecKiri);
}
else{
    digitalWrite(r2,HIGH);
    analogWrite (r1,0-kecKiri);
}
}

void tambahKiri(){
    lpdKiri++;
}
void tambahKanan(){
    lpdKanan++;
}

void waktuHabis(){
    nilaiLPDKiri = (1.201*lpdKiri - 36.872);
    nilaiLPDKanan =(1.201*lpdKanan - 36.872);
    errorKiri = targetKiri - nilaiLPDKiri;
    errorKanan = targetKanan - nilaiLPDKanan;
    Serial.println("=====KIRI=====");
    Serial.println(String(targetKiri)+","+String(nilaiLPDKiri)+","+String(pwm));
    Serial.println("=====KANAN=====");
    Serial.println(String(targetKanan)+","+String(nilaiLPDKanan)+","+String(pwm2));
    lpdKiri = 0;
    lpdKanan = 0;
}

void jalan(int mKiri, int mKanan){
    targetKiri = mKiri;
    targetKanan = mKanan;

    errorKiri = targetKiri - nilaiLPDKiri;
    if(errorKiri!=0){ I=I+errorKiri; }
    else {I=0; }
    pwm = (pwm + (errorKiri*kp))+ (errorKiri - lasterrorKiri)*kd + (ki*I)/35;
    lasterrorKiri=errorKiri;
    if(pwm>255)
        pwm=255;
    else if(pwm<0)pwm=0;

    errorKanan = targetKanan - nilaiLPDKanan;
```

```
if(errorKanan!=0){I2=I2+errorKanan;}
else {I2=0;}
pwm2 = (pwm2 + (errorKanan*kp2))+ (errorKanan - lasterrorKanan)*kd2 +
(ki2*I2)/35;
lasterrorKanan=errorKanan;
if(pwm2>255)
pwm2=255;
else if(pwm2<0)pwm2=0;

motor(pwm,pwm2);
}
void kirim(){
String cmd = "AT+CIPSTART=\"TCP\", \"184.106.153.149\",80\"; //ip
api.thingspeak.com
Serial3.println(cmd);
delay(500);
//coba kirim data
String getStr = "GET /update?api_key=";
getStr += WRITE_APIKEY;
getStr += "&field1=";
getStr += temp;//silahkan ganti nilai sensor
getStr += "\r\n\r\n";

//Serial.print("AT+CIPSEND=");
//Serial.println(cmd.length());
cmd = "AT+CIPSEND=";
cmd += String(getStr.length());

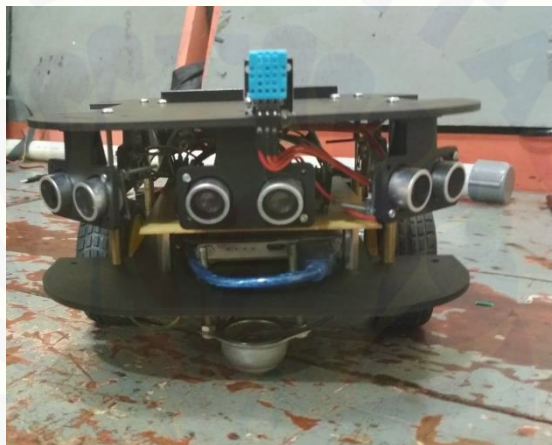
Serial3.println(cmd);

if(Serial3.find(">")){
Serial3.print(getStr);
Serial3.println(getStr);
}else{
Serial3.println("AT+CIPCLOSE");
}
}
//end coba kirim data
delay(500);
}
void dhtProg(){
err=dht11.read(humi, temp);
//delay(100); //delay for reread
if (temp >= 30){
digitalWrite(A0, HIGH);}
else{
digitalWrite(A0, LOW);
}
}
}
```

**B. Gambar Hardware**



Gambar Robot Tampak Samping

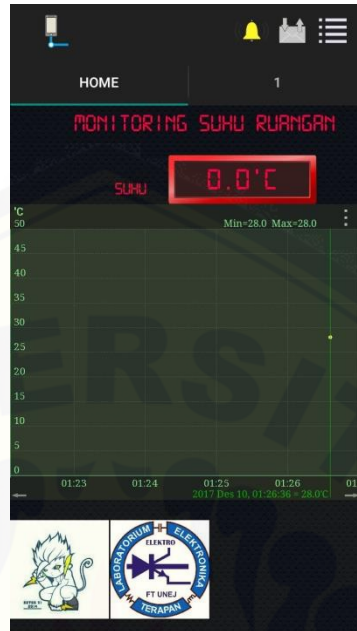


Gambar Robot Tampak Depan



Gambar Robot Tampak Atas

C. Gambar Software Aplikasi Android



Gambar aplikasi sebelum disambung IoT



Gambar aplikasi sesudah disambung IoT