

**SISTEM PENGENDALIAN KECEPATAN MOTOR PENDORONG ROBOT
HOVERCRAFT LINE FOLLOWER MENGGUNAKAN KONTROLER PID
BERBASIS MIKROKONTROLER ATMEGA 8535**

SKRIPSI

*Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik*



Disusun oleh:

ADECK APRILYAN KURNIAHADI

NIM. 0910630021 - 63

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

JURUSAN TEKNIK ELEKTRO

MALANG

2013

LEMBAR PERSETUJUAN

SISTEM PENGENDALIAN KECEPATAN MOTOR PENDORONG ROBOT
HOVERCRAFT LINE FOLLOWER MENGGUNAKAN KONTROLER PID
BERBASIS MIKROKONTROLER ATMEGA 8535

SKRIPSI

*Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik*



Disusun oleh:

ADECK APRILYAN KURNIAHADI

NIM. 0910630021 - 63

Telah diperiksa dan disetujui oleh

Dosen Pembimbing:

Pembimbing I

Pembimbing II

Ir. Purwanto, MT.

NIP. 19540424 198601 1 001

Dr. Ir. Erni Yudaningtvas, MT.

NIP. 19650913 199002 2 001

LEMBAR PENGESAHAN

**SISTEM PENGENDALIAN KECEPATAN MOTOR PENDORONG ROBOT
HOVERCRAFT LINE FOLLOWER MENGGUNAKAN KONTROLER PID
BERBASIS MIKROKONTROLER ATMEGA 8535**

SKRIPSI

Disusun oleh:

ADECK APRILYAN KURNIAHADI

NIM. 0910630021 - 63

Skripsi ini telah diuji dan dinyatakan lulus
pada tanggal 1 November 2013

Majelis Penguji:

Goegoes Dwi Nusantoro, ST., MT.
NIP. 19711013 200604 1 001

Ir. Retnowati, MT.
NIP. 19511224 198203 2 001

Akhmad Zainuri, ST., MT.
NIDN. 0020018401

Mengetahui

Ketua Jurusan Teknik Elektro

M. Aziz Muslim, ST., MT., Ph.D
NIP. 19741203 200012 1 001

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT, karena dengan rahmat, taufik, dan hidayah-Nya lah skripsi ini dapat diselesaikan. Shalawat serta salam tetap penulis curahkan kepada junjungan Rasulullah Muhammad SAW. Skripsi berjudul “Sistem Pengendalian Kecepatan Motor Pendorong Robot *Hovercraft Line Follower* Menggunakan Kontroler PID Berbasis Mikrokontroler ATMega 8535” ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Brawijaya.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

- Mama Setya Utami, S.Pd tercinta, atas nafas, segala nasehat, kasih sayang, perhatian, dan kesabarannya sebagai *single fighter* di dalam membesarkan dan mendidik penulis, serta telah banyak mendoakan kelancaran penulis hingga terselesaikannya skripsi ini dan salam hormat kepada Papa R. Broto Hadi Sutjahjo,
- Kakak-kakak penulis Dinda Novrischa Setyaningrum sekeluarga, adik penulis Astrie Septianing Anggarini, serta keluargaku tercinta atas kasih sayang, do'a dan motivasinya,
- Bapak Dr. Ir. Sholeh Hadi Pramono, MS selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya,
- Bapak Aziz Muslim, ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya,
- Bapak Prof. Ir. Budiono Mismail, MSEE., Ph.D., Ibu Asri Wulandari, ST., MT., dan Bapak Dwi Fadila Kurniawan, ST., MT selaku Dosen Pembimbing Akademik penulis atas segala nasehat dan bimbingan yang telah diberikan,
- Bapak Ir. Purwanto, MT selaku Ketua Kelompok Dosen Keahlian Teknik Kontrol Jurusan Teknik Elektro Universitas Brawijaya dan juga Dosen

Pembimbing I atas segala ilmu, bimbingan, nasehat, gagasan, ide, saran, motivasi, dan bantuan yang telah diberikan,

- Ibu Dr. Ir. Erni Yudaningsy, MT selaku Dosen Pembimbing II atas segala ilmu, bimbingan, nasehat, gagasan, ide, saran, motivasi, dan bantuan yang telah diberikan,
- Staff Recording Jurusan Teknik Elektro,
- Rekan-rekan pengerjaan skripsi di Laboratorium Sistem Kontrol dan Laboratorium Mekatronika dan Robotika, Doniarsyah Zen, Adityan, Lalu Irjan, Gladi, Sam Budi, Rizal, Aka, Eky, Juang, Rafi, Lintang, Bona, Risma, Firda, Doni, Akhbar, Gio,
- Seluruh Keluarga Besar Asisten dan Laboran mbak Eka Desiana H. I., ST Laboratorium Sistem Kontrol Jurusan Teknik Elektro atas segala masukan-masukannya,
- Seluruh Keluarga Besar Ampere 2009 yang selalu memberi pengalaman-pengalaman, saran-saran, dan dukungan-dukungan yang membangun atas pengerjaan skripsi ini,
- Seluruh Keluarga Besar Jurusan Teknik Elektro, senior dan junior serta semua pihak yang tidak mungkin untuk dicantumkan namanya satu per satu, terima kasih banyak atas bantuan dan dukungannya,

Penulis menyadari bahwa skripsi ini masih belum sempurna, oleh karena itu penulis mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Malang, 21 Oktober 2013

Penulis

DAFTAR ISI

KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
ABSTRAK	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	3
1.4 Tujuan.....	3
1.5 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA	5
2.1 <i>Hovercraft</i>	5
2.2 Robot <i>Line Follower</i>	6
2.3 Sensor Cahaya.....	6
2.4 Motor DC.....	9
2.4.1 Motor DC <i>Brushless</i>	9
2.4.2 Motor Servo	12
2.5 Kontroler.....	14
2.5.1 Kontroler Proporsional	15
2.5.2 Kontroler Integral	15
2.5.3 Kontroler Diferensial	16
2.5.4 Kontroler Proporsional Integral Diferensial (PID)	16
2.5.5 Metode Perancangan Kontroler Proporsional Integral Diferensial (PID) Menggunakan Metode Ziegler-Nichols	17
2.6 <i>Pulse Width Modulation</i> (PWM).....	21
2.7 Mikrokontroler.....	22
2.8 Mikrokontroler Atmel AVR ATmega 8535	23
2.9 Program CodeVision AVR	24
BAB III METODOLOGI PENELITIAN	27

3.1	Spesifikasi Alat	27
3.2	Perancangan dan Realisasi Pembuatan Alat	28
3.2.1	Perancangan Perangkat Keras dan Realisasi Pembuatan Alat.....	28
3.2.2	Perancangan dan Perhitungan Komponen yang akan Digunakan	28
3.2.3	Perancangan Perangkat Lunak.....	28
3.3	Pengujian Alat.....	28
3.4	Pengambilan Kesimpulan	30
BAB IV PERANCANGAN DAN PEMBUATAN ALAT		31
4.1	Spesifikasi Sistem	31
4.2	Diagram Blok Sistem.....	32
4.3	Perancangan Perangkat Keras.....	33
4.3.1	Perancangan Rangka Robot <i>Hovercraft Line Follower</i>	34
4.3.2	Rangkaian <i>Board</i> Mikrokontroler.....	34
4.3.3	<i>Electronic Speed Controller (ESC)</i>	35
4.3.4	Pemilihan Motor DC <i>Brushless</i> dan <i>Propeller</i>	36
4.3.5	Pemilihan Sistem Kendali Arah.....	39
4.4	Penentuan Nilai Penguatan Kontroler.....	40
4.5	Perancangan Perangkat Lunak.....	41
BAB V PENGUJIAN DAN ANALISIS		42
5.1	Pengujian Catu Tegangan Baterai.....	42
5.1.1	Peralatan Pengujian.....	42
5.1.2	Prosedur Pengujian	42
5.1.3	Hasil Pengujian.....	43
5.2	Pengujian Tegangan Regulator dan ESC.....	44
5.2.1	Peralatan Pengujian.....	44
5.2.2	Prosedur Pengujian	44
5.2.3	Hasil Pengujian.....	45
5.3	Pengujian Tegangan Keluaran Sensor <i>Photo Reflective</i>	47
5.3.1	Peralatan Pengujian.....	47
5.3.2	Prosedur Pengujian	47
5.3.3	Hasil Pengujian.....	47
5.4	Pengujian RPM dan <i>Thrust</i> Motor DC <i>Brushless Hover</i>	50

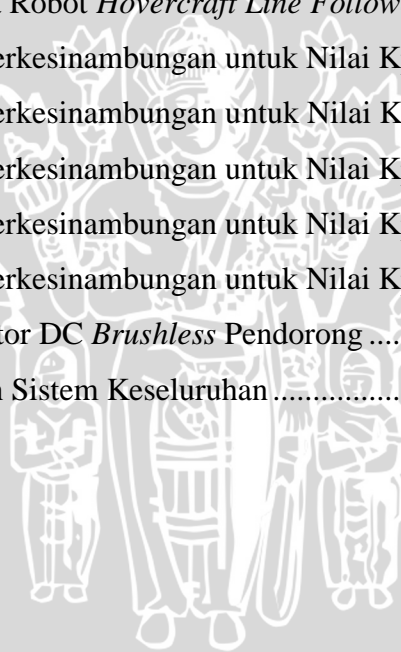


5.4.1	Peralatan Pengujian.....	50
5.4.2	Prosedur Pengujian	51
5.4.3	Hasil Pengujian	51
5.5	Pengujian Sinyal PWM Motor Servo	52
5.5.1	Peralatan Pengujian.....	52
5.5.2	Prosedur Pengujian	52
5.5.3	Hasil Pengujian	52
5.6	Pengujian Massa Robot <i>Hovercraft Line Follower</i>	54
5.6.1	Peralatan Pengujian.....	54
5.6.2	Prosedur Pengujian	54
5.6.3	Hasil Pengujian	54
5.7	Pengujian Nilai Kcr	56
5.7.1	Peralatan Pengujian.....	56
5.7.2	Prosedur Pengujian	56
5.7.3	Hasil Pengujian	56
5.8	Pengujian RPM Motor DC <i>Brushless Pendorong</i>	58
5.8.1	Peralatan Pengujian.....	58
5.8.2	Prosedur Pengujian	58
5.8.3	Hasil Pengujian	58
5.9	Pengujian Keseluruhan	59
5.9.1	Peralatan Pengujian.....	59
5.9.2	Prosedur Pengujian	60
5.9.3	Hasil Pengujian	60
BAB VI PENUTUP		63
6.1	Kesimpulan	63
6.2	Saran	63
DAFTAR PUSTAKA		65

DAFTAR GAMBAR

Gambar 2.1 Prinsip Kerja <i>Hovercraft</i>	6
Gambar 2.2 Rangkaian Sensor Detektor Garis.....	7
Gambar 2.3 Rangkaian Sensor Cahaya dengan Fototransistor.....	7
Gambar 2.4 Skematik Motor DC <i>Brushless</i>	10
Gambar 2.5 Komponen Motor DC <i>Brushless</i>	10
Gambar 2.6 Rangkaian Ekuivalen DC Motor.....	11
Gambar 2.7 Motor Servo.....	12
Gambar 2.8 Jalur Kabel Motor Servo.....	12
Gambar 2.9 Konfigurasi Pin Pengabelan Motor Servo.....	13
Gambar 2.10 Pengaturan Sudut Motor Servo.....	14
Gambar 2.11 Diagram Blok Kontroler Proporsional.....	15
Gambar 2.12 Diagram Blok Kontroler Integral.....	16
Gambar 2.13 Diagram Blok Kontroler Diferensial.....	16
Gambar 2.14 Diagram Blok Kontroler PID.....	17
Gambar 2.15 Kurva Respon Unit Step Menunjukkan 25% <i>Maximum Overshoot</i>	18
Gambar 2.16 Respon Plant Terhadap Masukan Berupa Unit Step.....	18
Gambar 2.17 Kurva yang Berbentuk S.....	18
Gambar 2.18 Sistem Loop Tertutup dengan Kontroler Proporsional.....	20
Gambar 2.19 Osilasi Berkesinambungan dengan Periode Pcr.....	20
Gambar 2.20 Sinyal PWM Secara Umum.....	21
Gambar 2.21 Konfigurasi Pin ATmega 8535 PDIP.....	24
Gambar 4.1 Diagram Blok Sistem secara Umum.....	32
Gambar 4.2 Diagram Blok Sistem Robot <i>Hovercraft Line Follower</i>	33
Gambar 4.3 Rangka Robot <i>Hovercraft Line Follower</i>	34
Gambar 4.4 Rangkaian <i>Board</i> Mikrokontroler.....	34
Gambar 4.5 <i>Electronic Speed Controller</i> 30A.....	36
Gambar 4.6 <i>Propeller</i> EMP 6x5E.....	38
Gambar 4.7 Motor DC <i>Brushless</i> 1400kV.....	39
Gambar 4.8 Motor Servo Toward Pro MG996R.....	39
Gambar 4.9 Respon Sistem.....	40

Gambar 4.10 <i>Flowchart</i> Perangkat Lunak.....	41
Gambar 5.1 Pengujian Catu Tegangan Baterai <i>Li-po</i> 11.1 V 2200 mA.....	43
Gambar 5.2 Grafik Catu Tegangan Baterai <i>Li-po</i> 11.1 V 2200 mA.....	43
Gambar 5.3 Pengujian Tegangan Regulator +5 V.....	45
Gambar 5.4 Pengujian Tegangan ESC.....	45
Gambar 5.5 Grafik Tegangan Regulator +5 V dan ESC.....	46
Gambar 5.6 Grafik Rata-rata Nilai Tegangan pada Logika Tiap Sensor.....	50
Gambar 5.7 Grafik <i>Thrust</i> Motor DC <i>Brushless Hover</i>	51
Gambar 5.8 Pengujian Sudut Motor Servo pada Sudut 110°	53
Gambar 5.9 Grafik <i>Duty Cycle</i> Motor Servo.....	53
Gambar 5.10 Pengujian Massa Robot <i>Hovercraft Line Follower</i> saat <i>OFF</i>	54
Gambar 5.11 Pengujian Massa Robot <i>Hovercraft Line Follower</i> saat <i>ON</i>	55
Gambar 5.12 Sinyal Osilasi Berkesinambungan untuk Nilai $K_p = 14$	56
Gambar 5.13 Sinyal Osilasi Berkesinambungan untuk Nilai $K_p = 16$	57
Gambar 5.14 Sinyal Osilasi Berkesinambungan untuk Nilai $K_p = 18$	57
Gambar 5.15 Sinyal Osilasi Berkesinambungan untuk Nilai $K_p = 20$	57
Gambar 5.16 Sinyal Osilasi Berkesinambungan untuk Nilai $K_p = 22$	57
Gambar 5.17 Grafik RPM Motor DC <i>Brushless</i> Pendorong.....	59
Gambar 5.12 Grafik Pengujian Sistem Keseluruhan.....	61



DAFTAR TABEL

Tabel 2.1 Kondisi Masukan terhadap Logika Keluaran	9
Tabel 2.2 Aturan Penalaran Ziegler-Nichols Berdasarkan Respon Unit Step dari Plan 19	
Tabel 2.3 Aturan Dasar Ziegler-Nichols Berdasarkan <i>Critical Gain</i> Kcr dan <i>Critical Period</i> Pcr	20
Tabel 4.1 Fungsi <i>Pin</i> Mikrokontroler	35
Tabel 4.2 Perhitungan <i>Thrust</i>	37
Tabel 4.3 Aturan Dasar Ziegler-Nichols Berdasarkan <i>Critical Gain</i> Kcr dan <i>Critical Period</i> Pcr	40
Tabel 5.1 Hasil Pengujian Catu Tegangan Baterai <i>Li-po</i> 11.1 V 2200 mA	43
Tabel 5.2 Hasil Pengujian Tegangan Regulator dan ESC	45
Tabel 5.3 Hasil Pengujian Sensor S0 Mengenai Garis Hitam	47
Tabel 5.4 Hasil Pengujian Sensor S1 Mengenai Garis Hitam	48
Tabel 5.5 Hasil Pengujian Sensor S2 Mengenai Garis Hitam	48
Tabel 5.6 Hasil Pengujian Sensor S3 Mengenai Garis Hitam	48
Tabel 5.7 Hasil Pengujian Sensor S4 Mengenai Garis Hitam	48
Tabel 5.8 Hasil Pengujian Sensor S5 Mengenai Garis Hitam	49
Tabel 5.9 Hasil Pengujian Sensor S6 Mengenai Garis Hitam	49
Tabel 5.10 Hasil Pengujian Sensor S7 Mengenai Garis Hitam	49
Tabel 5.11 Rata-rata Nilai Tegangan pada Logika Tiap Sensor	49
Tabel 5.12 Hasil Pengujian RPM dan <i>Thrust</i> Motor DC <i>Brushless Hover</i>	51
Tabel 5.13 Pengujian RPM Motor DC <i>Brushless</i> Pendorong	58
Tabel 5.14 Hasil Pengujian Sistem Keseluruhan.....	60

ABSTRAK

Adeck Aprilyan Kurniahadi, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, November 2013, Sistem Pengendalian Kecepatan Motor Pendorong Robot *Hovercraft Line Follower* Menggunakan Kontroler PID Berbasis Mikrokontroler ATmega 8535, Dosen Pembimbing: Ir. Purwanto, MT., Dr. Ir. Erni Yudaningtyas, MT.

Konstruksi *hovercraft* sangat dibutuhkan dalam menangani bencana yang memiliki jalur berlumpur dan berawa. Namun di Indonesia pengembangan kendaraan yang memiliki kelebihan jelajah ini masih sangat minim. Dalam skripsi ini *hovercraft* diaplikasikan dalam bentuk robot otomatis *line follower*. *Hovercraft line follower* adalah robot yang mampu bergerak mandiri dengan mengikuti garis sebagai *setpoint*. Pengendalian dirancang agar posisi sensor robot dapat tetap mengikuti garis dengan kecepatan motor pendorong yang dikendalikan.

Hasil pengujian dengan metode kedua Ziegler-Nichols didapatkan parameter PID yaitu $K_p = 12$, $K_i = 6.897$, dan $K_d = 5.22$. Hasil pengujian terhadap keseluruhan sistem robot *hovercraft line follower* dengan nilai parameter PID saat melaju mengikuti garis hitam dengan panjang 1 meter dan lebar 4.8 cm diperoleh *error* pergeseran terjauh sebesar 21.43 % (3 cm dari titik tengah sensor) dan *error* pergeseran terdekat sebesar 3.57 % (0.5 cm dari titik tengah sensor). Hasil ini menunjukkan robot dapat mengikuti garis dengan posisi sensor tidak lepas dari jalur dan pengendalian kecepatan motor pendorong sesuai dengan yang dibutuhkan sistem, namun tidak adanya *center gravity* yang baik pada rangka robot *hovercraft line follower* menyebabkan *error* pergeseran yang sangat besar.

Kata Kunci: *Hovercraft line follower*, Pengendalian kecepatan, PID, posisi robot.

BAB I PENDAHULUAN

1.1 Latar Belakang

Dewasa ini hampir tidak ada yang tidak mengenal robot. Namun pengertian robot tidak dipahami secara sama oleh setiap orang. Istilah robot berasal dari bahasa Cheko “robota” yang berarti pekerja atau kuli yang tidak mengenal lelah atau bosan. Robot merupakan sebuah alat mekanik yang dapat memperoleh informasi. Informasi dari lingkungan (melalui sensor), dapat diprogram, dapat melaksanakan beberapa tugas yang berbeda, bekerja secara otomatis, dan cerdas (kecerdasan buatan). Robot biasa digunakan untuk tugas yang berat, berbahaya, dan pekerjaan berulang, pembersih limbah beracun, penjelajah bawah air dan luar angkasa, pertambangan, pekerjaan “cari dan tolong” (*search and rescue*), pencarian tambang, hiburan, sampai alat bantu rumah tangga (Taufiq, 2010: 1, 2).

Robot memiliki berbagai macam konstruksi, antara lain: robot mobil (*mobile robot*), robot manipulator (tangan), robot *humanoid*, *flying robot*, robot berkaki, robot jaringan, robot animalia, dan robot *cyborg*. Robot mobil (*mobile robot*) dapat berpindah posisi dari satu titik ke titik yang lain. Robot manipulator (tangan) dapat bekerja menyerupai lengan manusia. Robot *humanoid* menyerupai manusia secara utuh. *Flying robot* dapat terbang seperti pesawat. Robot berkaki dapat berpindah posisi dari satu titik ke titik lain. Robot jaringan yang dikendalikan melalui internet dengan protocol TCP/IP. Robot animalia yang menyerupai hewan secara utuh. Robot *cyborg* dengan penggabungan bagian organik dan bagian *cybernetic* (Taufiq, 2010: 2, 3).

Dari macam konstruksi yang ada, *mobile robot* sangat disukai bagi orang yang mulai mempelajari robot. Dalam mempelajari *mobile robot* tidak memerlukan kerja fisik yang berat dan minimal hanya diperlukan pengetahuan tentang mikrokontroler dan sensor-sensor elektronik. *Mobile robot* dapat dibuat sebagai pengikut garis (*Line Follower*) atau pengikut dinding (*Wall Follower*) ataupun pengikut cahaya. Namun, *mobile robot* dengan ciri khas aktuator berupa roda untuk menggerakkan keseluruhan badan robot masih memiliki kekurangan. Terutama pada medan-medan sulit untuk berfungsi dalam tugas “cari dan tolong” (*search and rescue*) yaitu medan berair atau

berlumpur. Dalam medan tersebut dapat dilakukan dengan baik oleh jenis kendaraan berbentuk *hovercraft*.

Hovercraft merupakan jenis kendaraan yang memanfaatkan gaya angkat dan gaya dorong. Gaya angkatnya tidak membutuhkan landasan luncur dan dapat tetap dalam keadaan melayang pada angkatan penuh. Gaya dorong yang dibutuhkan lebih rendah dari gaya angkatnya. Keadaan melayang menjadi alasan *hovercraft* dapat bergerak hampir di semua medan. Dapat bergerak di atas semua permukaan (berair, berlumpur, maupun bidang datar) dan gesekan yang ditimbulkan lebih kecil daripada kendaraan darat maupun kapal (Hendra, 2010: 2, 3).

Dalam kaitannya tugas *mobile robot* "cari dan tolong" (*search and rescue*) dan beberapa kelemahannya melalui beberapa medan, maka dirancanglah pengembangan dari sebuah *mobile robot*. Dengan mengganti bentuk konstruksi *mobile robot* beroda ke bentuk konstruksi *hovercraft* yang difungsikan sebagai robot *line follower*.

Proporsional Integral Diferensial (PID) adalah kontroler yang merupakan gabungan dari kontroler proposional, kontroler integral dan kontroler differensial. Gabungan dari ketiga kontroler ini diharapkan agar mendapat keluaran sistem yang sesuai untuk konstruksi robot *hovercraft line follower*. Kontroler PID ditanamkan untuk mengendalikan kecepatan motor pendorong robot *hovercraft line follower*. Keuntungan dari kontroler PID adalah merupakan sebuah sistem yang sederhana sehingga lebih cepat dalam mengambil sebuah keputusan.

Diharapkan dengan menggunakan kontroler PID respon perubahan kecepatan motor pendorong cepat dalam mengikuti garis sebagai *setpoint*, *offset* yang terjadi kecil, dan *overshoot* yang terjadi kecil. Sehingga hasilnya dapat dijadikan referensi atau kontribusi ilmiah untuk membangun *hovercraft-hovercraft* yang lebih cerdas dan dapat diwujudkan sebagai kendaraan "cari dan tolong" (*search and rescue*) tanpa awak pengemudi yang canggih.

1.2 Rumusan Masalah

1. Bagaimana merancang *hardware* sistem pengendalian kecepatan motor pendorong pada robot *hovercraft line follower* menggunakan kontroler PID berbasis mikrokontroler ATmega 8535?

2. Bagaimana merancang *software* sistem pengendalian kecepatan motor pendorong robot *hovercraft line follower* dengan menggunakan kontroler PID berbasis mikrokontroler ATmega 8535?

1.3 Batasan Masalah

Untuk menekankan pada objek pembahasan yang ada, maka pada penelitian ini diberikan batasan masalah sebagai berikut:

1. Robot *hovercraft line follower* yang digunakan adalah miniatur yang dibuat dengan desain sendiri. Dengan massa 1.506 kg dan dimensi panjang 45 cm, lebar 28 cm, dan tinggi 24 cm.
2. Digunakan 5 motor DC yaitu 2 motor jenis motor DC *brushless* C2822-1400kV untuk menghasilkan gaya dorong dan gaya angkat robot dan 1 jenis motor servo Toward Pro MG996R serta 2 motor DC magnet permanen untuk kendali arah.
3. Pembahasan ditekankan pada penggunaan kontroler PID pada robot *hovercraft line follower*.
4. Pengendalian yang dilakukan adalah pengendalian kecepatan putaran motor DC *brushless* pada motor kipas pendorong saat mengikuti garis hitam.
5. Sensor yang digunakan adalah sensor cahaya dengan *photo reflective*.

1.4 Tujuan

Merancang *hardware* dan *software* robot *hovercraft line follower* dengan sistem pengendalian kecepatan motor pendorong menggunakan kontroler PID.

1.5 Sistematika Penulisan

Agar penyusunan laporan skripsi ini dapat mencapai sasaran dan tidak menyimpang dari judul yang telah ditentukan, maka diperlukan sistematika pembahasan yang jelas. Pembahasan dalam skripsi ini secara garis besar adalah sebagai berikut:

BAB I Pendahuluan

Menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB II Tinjauan Pustaka

Menjelaskan teori dasar yang berisi penjelasan tentang teori *hovercraft*, robot *line follower*, sensor cahaya, motor DC *brushless*, motor DC servo, kontroler, kontroler PID, dan rangkaian mikrokontroler.

BAB III Metodologi Penelitian

Menjelaskan tentang metodologi penelitian yang terdiri atas studi literatur, perancangan alat, pembuatan alat, pengujian alat, serta pengambilan kesimpulan dan saran.

BAB IV Perancangan dan Pembuatan Alat

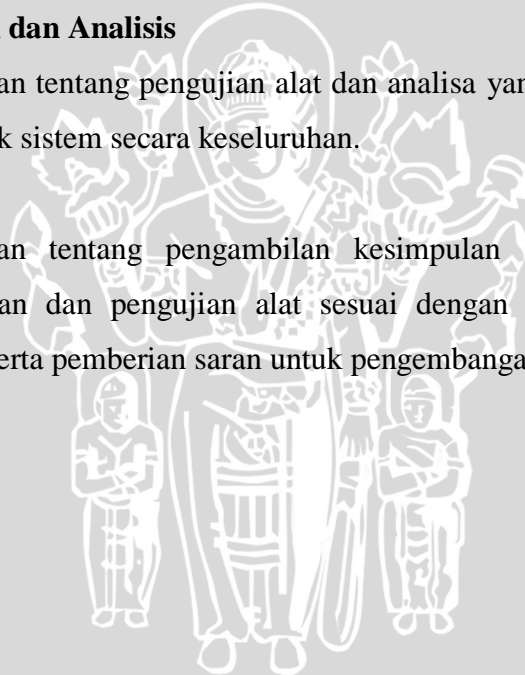
Menjelaskan tentang perancangan dan pembuatan alat yang meliputi prinsip kerja alat, perancangan perangkat keras dan perangkat lunak.

BAB V Pengujian dan Analisis

Menjelaskan tentang pengujian alat dan analisa yang meliputi pengujian bagian blok sistem secara keseluruhan.

BAB VI Penutup

Menjelaskan tentang pengambilan kesimpulan sesuai dengan hasil perealisasiian dan pengujian alat sesuai dengan tujuan dan rumusan masalah, serta pemberian saran untuk pengembangan.



BAB II

TINJAUAN PUSTAKA

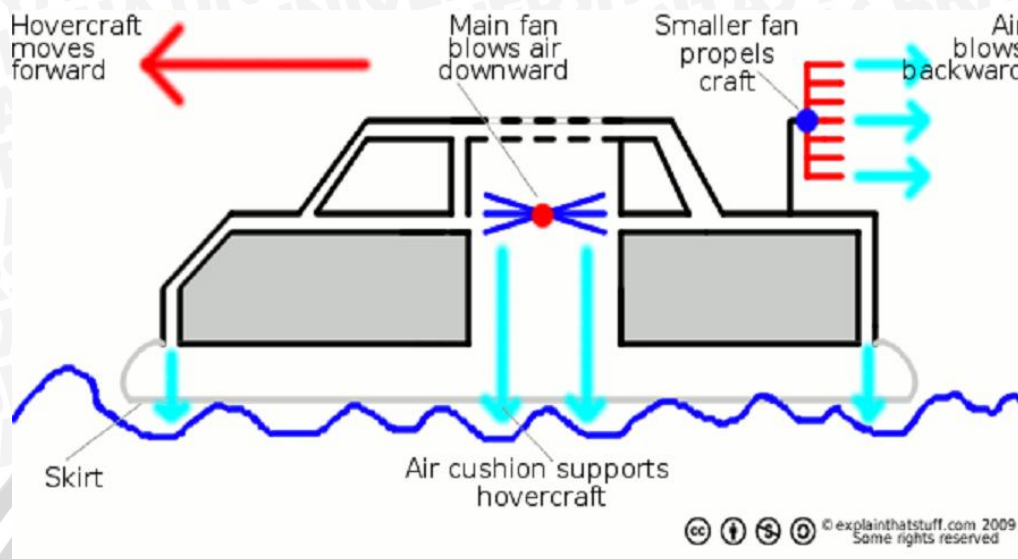
2.1 *Hovercraft*

Kapal bantalan udara atau *hovercraft* adalah suatu kendaraan yang berjalan di atas bantalan udara (*air cushion*). Bantalan udara tersebut ditimbulkan dengan cara meniupkan udara ke ruang bawah kapal (*plenum chamber*). Udara ditahan oleh *skirt* (sekat lentur) mengakibatkan tekanan udara di dalam *plenum chamber* lebih tinggi daripada tekanan udara luar dan timbul gaya angkat. Untuk menggerakkan *hovercraft*, digunakan gaya dorong yang diperoleh dari baling-baling seperti pada pesawat udara. Gaya angkat *hovercraft* bekerja pada penampang yang luas. Sehingga tekanan terhadap tanah atau air (*ground pressure*) yang ditimbulkan kecil. Dengan demikian, *hovercraft* dapat berjalan di atas lumpur, air maupun daratan dengan membawa beban yang cukup berat. Karena tidak adanya kontak langsung antara *hovercraft* dan permukaan daratan atau air, maka hambatan yang terjadi kecil sehingga *hovercraft* dapat melaju dengan kecepatan tinggi (Hendra, 2010: 3, 4).

Komponen utama penyusun *hovercraft* antara lain:

- Lambung (*hull*) yaitu badan kapal yang dapat dibuat dari aluminium ataupun serat kaca yang dibuat kedap air. Rongga di dalam lambung ini diisi dengan busa poliuretana yang membuat *hovercraft* tetap mengapung jika terjadi kebocoran pada lambung.
- *Skirt* yaitu bagian *hovercraft* yang berfungsi untuk menahan udara di bawah *hovercraft* agar tidak mudah keluar. *Skirt* terbuat dari tekstil yang dilapisi karet untuk menjaga agar udara tetap berada di dalam ruang di bawah lambung kapal.
- sumber tenaga *hovercraft* biasanya disediakan oleh mesin diesel atau bensin. Mesin digunakan untuk memutar baling-baling yang akan menghasilkan gaya dorong.

Prinsip kerja hovercraft diperlihatkan dalam Gambar 2.1.



Gambar 2.1 Prinsip Kerja *Hovercraft* (Fajar Prihatama, 2011: 1)

Ketika *main fan* bekerja menghasilkan tiupan udara kebagian bawah lambung, maka udara akan tertahan oleh *skirt* dan mengakibatkan timbul gaya angkat akibat perubahan tekanan yang terjadi. Pada saat inilah dikatakan *hovercraft* pada posisi melayang. Ketika *smaller fan* bekerja menghasilkan tiupan udara kebelakang, maka udara akan menghasilkan gaya dorong pada *hovercraft* sehingga dapat bergerak maju.

2.2 Robot *Line Follower*

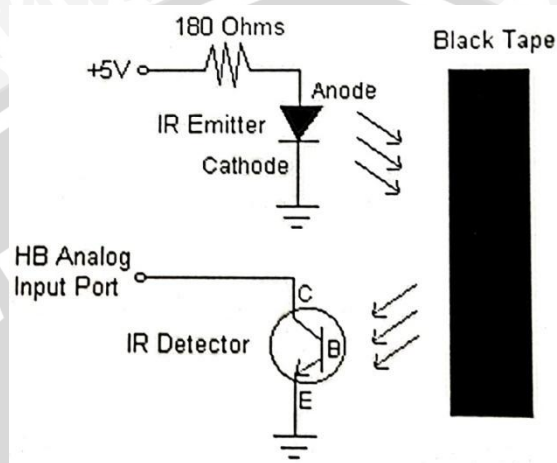
Robot *line follower* adalah robot yang bergerak secara otomatis mengikuti lika-liku garis di permukaan lantai. Robot dapat mengenali pola garis, menggunakan sensor cahaya. Sensor cahaya mendeteksi terang gelap permukaan yang ada di bawah sensor dengan memanfaatkan pantulan balik cahaya yang dikirim. Terang gelap permukaan harus diperoleh perbedaan warna yang mencolok antara warna garis dan warna permukaan lantai.

2.3 Sensor Cahaya

Sensor yang digunakan termasuk dalam jenis sensor fototransistor infra merah. Sensor ini terdiri atas LED infra merah dan fototransistor sebagai penerima cahaya pantul dari infra merah dengan panjang gelombang cahaya yang dipancarkan 950 nm (VISHAY, 2011: 1). Sensor fototransistor infra merah dapat digunakan untuk

menghasilkan sinyal listrik. Besar sinyal listrik yang dihasilkan tergantung dari jumlah cahaya yang mengenainya dalam rentang 750 nm – 1500 nm untuk jarak dekat. Semakin kuat cahaya, semakin besar arus balik. Secara umum, besarnya arus balik fototransistor infra merah adalah satuan miliampere (Taufiq, 2010: 221).

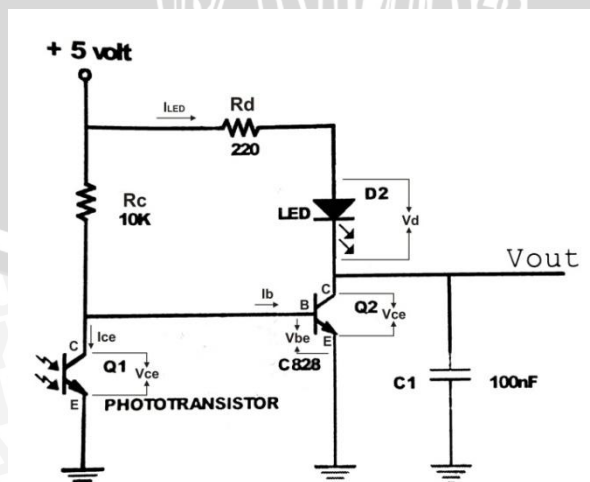
Rangkaian sensor detektor garis terlihat dalam Gambar 2.2.



Gambar 2.2 Rangkaian Sensor Detektor Garis (Taufiq, 2010: 218)

Rangkaian *Infra Red (IR) Emitter* akan memancarkan cahaya infra merah pada permukaan. Banyak sedikitnya cahaya pantulan yang akan diterima *Infra Red (IR) Detector* bergantung pada gelap terang permukaan. Semakin gelap permukaan maka pantulan cahaya yang diterima semakin sedikit dan sebaliknya, semakin terang permukaan maka pantulan cahaya yang diterima semakin banyak.

Untuk rangkaian sensor cahaya dengan fototransistor diperlihatkan dalam Gambar 2.3.



Gambar 2.3 Rangkaian Sensor Cahaya dengan Fototransistor (Taufiq, 2010: 219)

Dalam Gambar 2.3, arus yang mengalir dalam terminal kolektor menuju *emitter* (Ice) saat fototransistor Q1 mengalami saturasi (ada berkas infra merah yang diterima) dibatasi oleh resistor 10k Ω sehingga

$$I_{ce} = \frac{(V_{cc}-V_{ce})}{R_c} \dots\dots\dots(2.1)$$

- dengan :
- I_{ce} = arus yang mengalir dari *collector* ke *emitter* (mA)
 - V_{cc} = tegangan catuan 5 V (V)
 - V_{ce} = tegangan dari *collector* ke *emitter* (V)
 - R_c = resistansi *collector* Q1 (Ω)

Jadi,

$$I_{ce} = \frac{(V_{cc}-V_{ce})}{R_c} = \frac{(5-0)\text{volt}}{10\text{ k}} = 0,5\text{ mA}$$

Saat Q1 tersaturasi, Ib dalam Q2 akan mendekati 0 (dengan menganggap $V_{ceQ1} \approx V_{beQ2}$) yang mengakibatkan Q2 dalam kondisi putus sehingga keluaran dalam terminal kolektor (V_{ceQ2}) akan sebanding dengan $V_{cc} = 5$ volt, dalam ambang logika digital $V_{out} \approx 5$ volt \approx HIGH.

Fototransistor Q1 berada dalam kondisi putus ($I_{ce} \approx 0$) saat tidak ada berkas inframerah yang diterima sehingga $V_{ceQ1} \approx V_{beQ2} \approx 0,6$ volt dan terjadi lonjakan Ib dalam Q2 dari 0 hingga sebesar

$$I_b = \frac{(V_{cc}-V_{be})}{R_c} = \frac{(5-0,6)\text{volt}}{10\text{ k}} = 0,44\text{ mA}$$

dengan : I_b = arus yang mengalir pada kaki base Q2 (mA)

Kondisi tersebut akan mengakibatkan Q2 tersaturasi dan keluaran $V_{ceQ2} \approx 0$, dalam ambang logika digital $V_{out} \approx 0 \approx$ LOW, dan akan diikuti oleh beda potensial pada electrode LED dengan tegangan maju jatuh V_D (*forward bias*) sebesar 1,6 volt hingga 2,2 volt (untuk warna merah) sehingga besarnya arus yang dialirkan oleh diode sampai mengeluarkan cahaya dengan spectrum merah (diasumsikan $V_D = 2$ volt) adalah

$$I_{LED} = \frac{(V_{cc}-V_d)}{R_d} = \frac{(5-2)\text{volt}}{220} = 13,6\text{ mA}$$

Arus yang dibutuhkan LED untuk menyala maksimum sebesar 10 mA sampai dengan 20 mA sehingga nilai arus (I_{LED}) yang dibatasi oleh R_d telah optimal untuk memberikan pancaran cahaya indikasi. Tabel 2.1 memberikan kesimpulan karakteristik sistem masukannya (*sensing*) (Taufiq, 2010: 219, 220).

Kondisi masukan terhadap logika keluaran ditunjukkan dalam Tabel 2.1.

Tabel 2.1 Kondisi Masukan terhadap Logika Keluaran (Taufiq, 2010: 220)

Berkas IR Diterima Sensor	Sistem Keluaran
ADA	HIGH (LED off)
KOSONG	LOW (LED on)

2.4 Motor DC

Motor akan memiliki tiga fungsi, yaitu untuk penghasil udara yang dihembuskan ke dalam bantalan udara yang akan mengangkat keseluruhan badan *hovercraft*, untuk penghasil udara yang dihembuskan sebagai pendorong keseluruhan badan *hovercraft*, dan sebagai pengarah hembusan angin sebagai pendorong *hovercraft*. Motor yang digunakan terdapat tiga jenis yaitu motor servo, motor DC magnet permanen, dan motor DC *brushless*.

2.4.1 Motor DC *Brushless*

Motor DC *brushless* atau juga dikenal sebagai motor komutasi elektrik adalah motor elektrik yang dicatu dengan arus/ tegangan DC (*direct-current*) melalui inverter terintegrasi/ *switching power supply*. Inverter terintegrasi menghasilkan sinyal listrik AC ke driver motor (arus bi-directional dengan tidak ada pembatasan pada gelombang) dan mempunyai sistem komutasi elektrik, dibandingkan dengan komutator mekanik dengan sikat. Hubungan antara arus dan torsi serta frekuensi dan kecepatan adalah linier (Brown, 2002: 6)

$$M = K_T \times I_{line} \dots\dots\dots(2.2)$$

dengan: I_{line} = Arus tiap jalur (A)
 M = torsi (N.m)
 K_T = konstanta torsi BLDC (0.00083)

$$RPM = K_v \times V \dots\dots\dots(2.3)$$

dengan: RPM = kecepatan putaran per menit
 K_v = konstanta BLDC (1400)
 V = tegangan masukan

Jika RPM dikonversi dalam satuan kecepatan anguler (ω) maka didapatkan:

$$\omega = \left(\frac{RPM}{60}\right) \times 2\pi \dots\dots\dots(2.4)$$

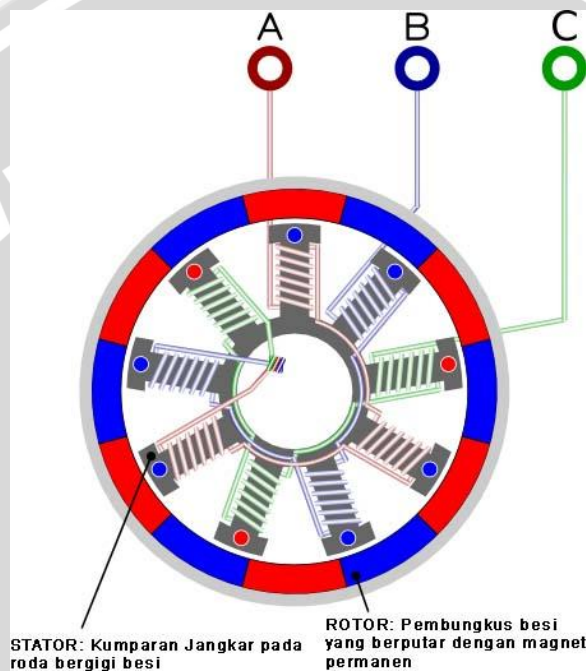
dengan: ω = kecepatan anguler (rad/s)
 π = 3.14



RPM = kecepatan putaran per menit

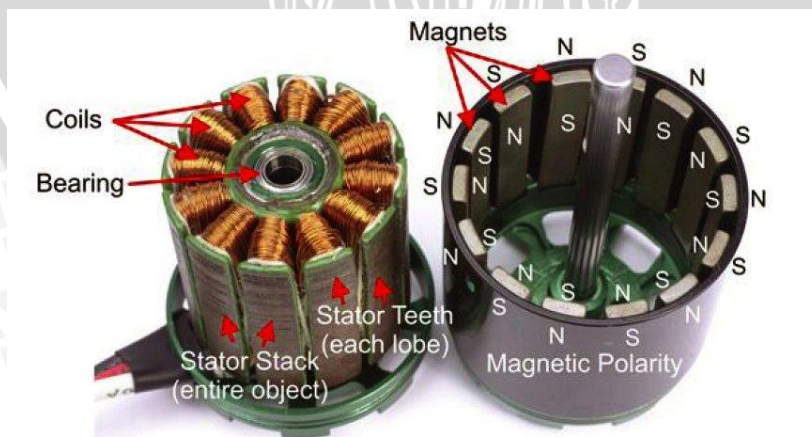
Motor DC *brushless* juga bisa dideskripsikan sebagai motor *stepper*. Motor DC *brushless* memiliki magnet permanen yang tetap dan mungkin juga dengan beberapa pole pada rotor daripada stator. Versi yang lain mungkin tanpa magnet permanen, hanya pole yang diinduksi pada rotor kemudian ditarik pada lilitan stator.

Skematik motor DC *brushless* diperlihatkan dalam Gambar 2.4.



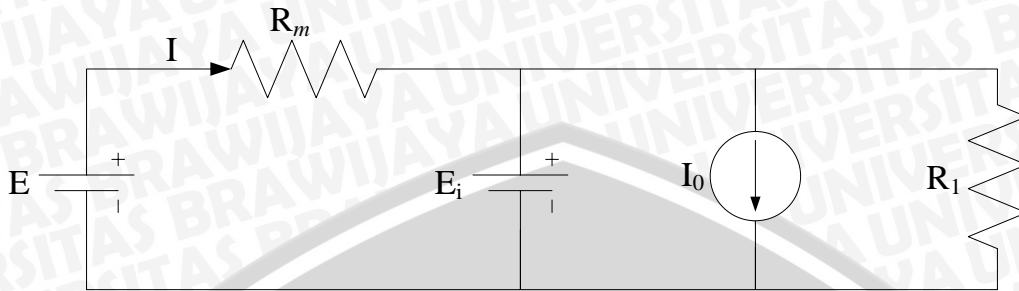
Gambar 2.4 Skematik Motor DC *Brushless* (Buchi, 2012: 12)

Komponen motor DC *brushless* yang ditunjukkan dalam Gambar 2.5.



Gambar 2.5 Komponen Motor DC *Brushless* (www.rcuniverse.com)

Rangkaian ekivalen DC motor ditunjukkan dalam Gambar 2.6.



Gambar 2.6 Rangkaian Ekivalen DC Motor (Brown, 2002: 6)

Sesuai dengan rangkaian ekivalen motor DC *brushless* dalam Gambar 2.6, dapat ditentukan hubungan (Brown, 2002: 6):

$$E_i = E - IR_m \dots\dots\dots (2.5)$$

- dengan :
- E_i = *back*-EMF (V)
 - E = tegangan baterai (V)
 - I = arus baterai (A)
 - R_m = resistansi *winding* (Ω)

Maka kecepatan putaran motor ditentukan melalui:

$$K_v = \frac{N}{E_i} \dots\dots\dots (2.6)$$

- dengan :
- K_v = konstanta RPM/ volt
 - N = kecepatan putaran motor (RPM)
 - E_i = *back*-EMF (V)

nilai R_1 dari rangkaian ekivalen motor BLDC dapat ditentukan melalui:

$$R_1 = \frac{E - I(R_m + R_{ESC})}{I - I_0} \dots\dots\dots (2.7)$$

- dengan :
- R_1 = rugi arus Eddy (Ω)

Seperti motor 3 fasa pada umumnya, motor DC *brushless* memiliki 2 jenis lilitan pada statornya yaitu *star* (Y) dan *delta* (Δ). Hubungan pada nilai torsi antara keduanya ditunjukkan dengan persamaan (Brown, 2002: 6):

$$T_Y = \sqrt{T_\Delta} \dots\dots\dots (2.8)$$

- dengan :
- T = torsi motor DC *brushless* (N.m)

$$\omega_Y = \frac{1}{\sqrt{3}} \omega_\Delta \dots\dots\dots (2.9)$$

- dengan :
- ω = kecepatan anguler (rad/s)

2.4.2 Motor Servo

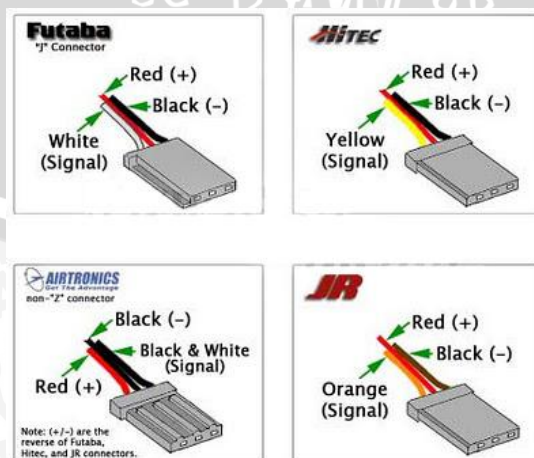
Berbeda dengan motor DC dan motor *Stepper*, motor servo adalah motor dengan sistem *closed feedback* yang berarti posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada dalam motor servo. Motor ini terdiri atas sebuah motor, serangkaian *internal gear*, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut putaran servo. Sedangkan sudut sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor.

Motor servo mampu bekerja dua arah yaitu *clockwise* (CW) atau searah jarum jam dan *counter clockwise* (CCW) atau berlawanan arah jarum jam yang arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal pada bagian pin kontrolnya. Gambar 2.7 menunjukkan gambar motor servo.



Gambar 2.7 Motor Servo (Parallax Inc. Ver. 1.1: 1)

Beberapa jalur kabel motor servo diperlihatkan dalam Gambar 2.8.

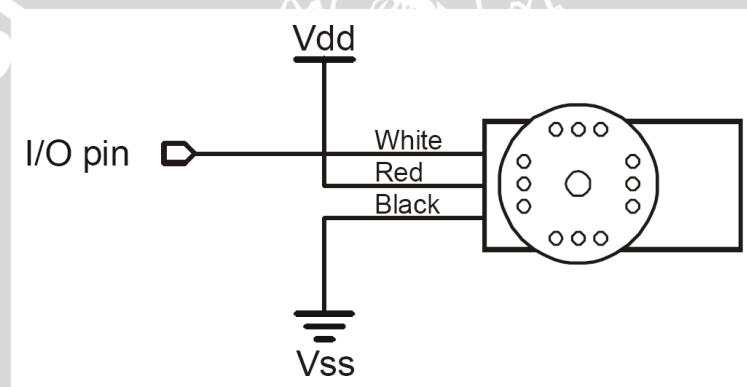


Gambar 2.8 Jalur Kabel Motor Servo (Sawicz, 2003: 1)

Motor servo merupakan motor yang berputar lambat, biasanya ditunjukkan oleh *rate* putarannya yang lambat, namun demikian memiliki torsi yang kuat karena *internal gear*-nya. Karakteristik motor servo adalah sebagai berikut:

1. Memiliki 3 jalur kabel: *power*, *ground*, dan *control* seperti ditunjukkan dalam Gambar 2.8.
2. Pin *Control* untuk mengendalikan posisi.
3. Operasional dari motor servo dikendalikan oleh sebuah pulsa selebar ± 20 ms, dengan lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari *range* sudut maksimum.
4. Konstruksi didalamnya meliputi *internal gear*, *potensiometer*, dan *feedback control*.

Gambar 2.9 menunjukkan konfigurasi pin pengabelan motor servo.



Gambar 2.9 Konfigurasi Pin Pengabelan Motor Servo (Parallax Inc. Ver. 1.1: 2)

Secara umum terdapat dua jenis motor servo, yaitu:

1. Motor Servo *Standar 180°*

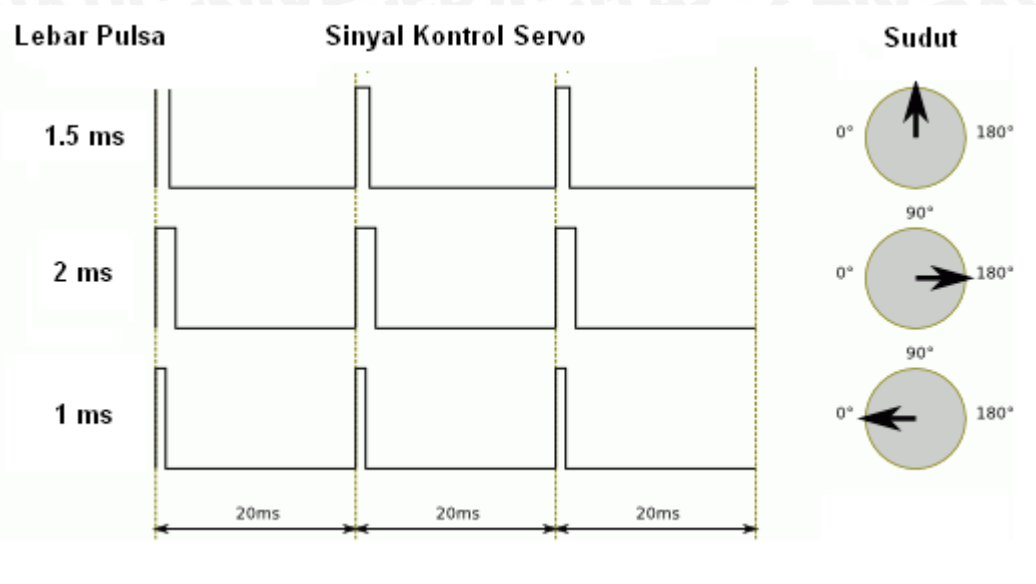
Motor servo jenis ini merupakan motor yang hanya mampu bergerak dua arah *Clockwise* (CW) dan *Counter Clockwise* (CCW) dan mempunyai *defleksi* masing-masing sudut mencapai 90° sehingga total *defleksi* sudut dari kanan – tengah – kiri adalah 180° .

2. Motor Servo *Continuous*

Motor servo jenis ini mampu bergerak dua arah *Clockwise* (CW) dan *Counter Clockwise* (CCW) dan tanpa batasan *defleksi* sudut putar (dapat berputar secara kontinyu) sehingga motor ini berputar 360° .

Pengaturan sudut motor servo diperlukan untuk mengetahui gerakan dari motor servo dengan pulsa yang harus diberikan untuk bergerak ke kanan atau bergerak ke kiri.

Teknik pengaturan sudut motor servo ditunjukkan Gambar 2.10.



Gambar 2.10 Pengaturan Sudut Motor Servo (Parallax Inc. Ver. 2.1, 2010: 4)

2.5 Kontroler

Sistem pengendalian dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengendalian adalah harus stabil. Disamping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif, yakni tolok ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikenai gangguan (Ogata K.,1997). Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam merespons *input*, dan bagaimana peredaman terhadap adanya lonjakan (*over shoot*).

Suatu sistem dikatakan stabil jika diberi gangguan maka sistem tersebut akan kembali ke keadaan *steady state* di mana *output* berada dalam keadaan tetap seperti tidak ada gangguan. Sistem dikatakan tidak stabil jika *output*nya berosilasi terus menerus ketika dikenai suatu gangguan. Karena suatu sistem pengendalian biasanya melibatkan penyimpanan energi maka *output* sistem ketika diberi suatu *input*, tidak dapat mengikuti *input* secara serentak, tapi menunjukkan respons transien berupa suatu osilasi teredam sebelum mencapai *steady state*.

Prinsip kerja kontroler adalah membandingkan nilai aktual keluaran plan dengan nilai referensi. Kemudian kontroler menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan kesalahan (Ogata, 1997).

2.5.1 Kontroler Proporsional

Untuk kontroler dengan aksi kontrol proporsional, hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ adalah:

$$m(t) = K_p e(t) \dots\dots\dots (2.10)$$

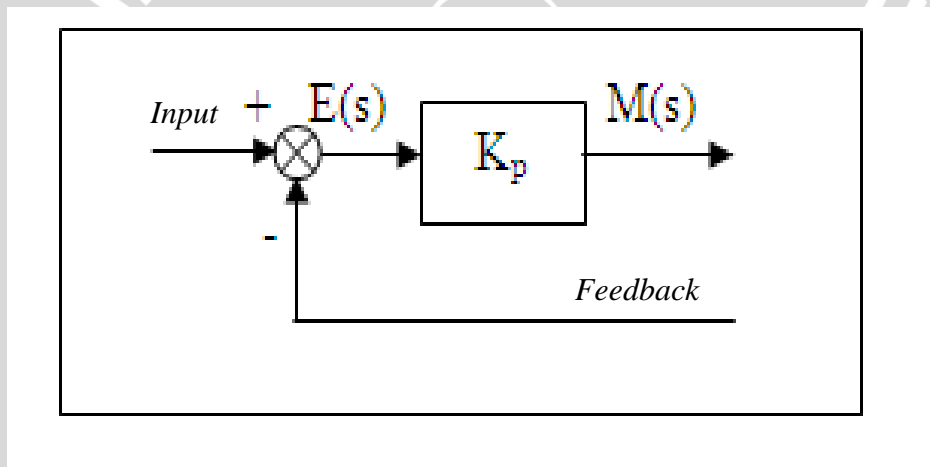
atau, dalam besaran transformasi Laplace,

$$\frac{M(s)}{E(s)} = K_p \dots\dots\dots (2.11)$$

di mana K_p adalah kepekaan proporsional atau penguatan.

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggerakannya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K.,1997).

Diagram blok kontroler proporsional ditunjukkan dalam Gambar 2.11.



Gambar 2.11 Diagram Blok Kontroler Proporsional (Ogata K., 1997)

2.5.2 Kontroler Integral

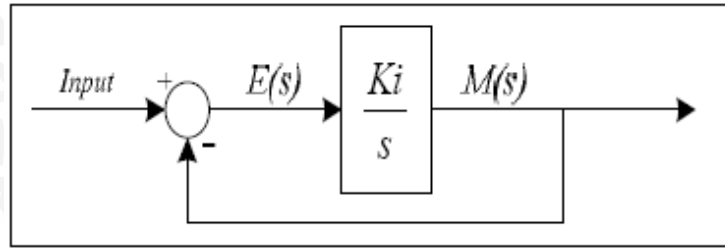
Pada kontroler dengan aksi integral, harga keluaran kontroler $m(t)$ diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak $e(t)$.

Jadi,

$$\frac{dm(t)}{dt} = K_i e(t) \dots\dots\dots (2.12)$$

dengan K_i adalah konstanta integral. Jika harga $e(t)$ diduakalikan, maka harga $m(t)$ berubah dengan laju perubahan menjadi dua kali semula. Jika kesalahan penggerak nol, maka harga $m(t)$ tetap stasioner. Aksi kontrol integral seringkali disebut kontrol *reset* yang digunakan untuk menghilangkan *error steady state* (Ogata K.,1997).

Diagram blok kontroler integral ditunjukkan dalam Gambar 2.12.



Gambar 2.12 Diagram Blok Kontroler Integral (Ogata K., 1997)

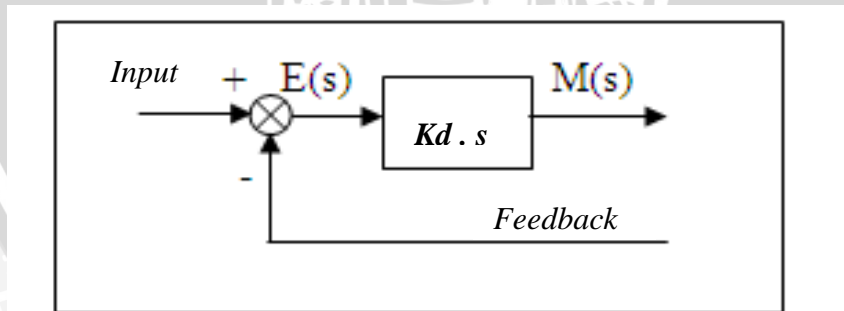
2.5.3 Kontroler Diferensial

Kontroler ini digunakan untuk memperbaiki atau mempercepat respons transien sebuah sistem kontrol dengan cara memperbesar *phase lead* terhadap penguatan kontrol dan mengurangi *phase lag* penguatan tersebut (Ogata K.,1997). Kontroler diferensial tidak dapat mengeluarkan *output* bila tidak ada perubahan *input*, selain itu kontroler differensial tidak dapat digunakan untuk proses yang mengandung *noise*. Hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ adalah :

$$\frac{M(s)}{E(s)} = Kd \cdot s \dots\dots\dots (2.13)$$

dengan Kd adalah konstanta diferensial yang digunakan untuk memperbaiki atau mempercepat respons transien sebuah sistem serta dapat meredam osilasi.

Gambar 2.13 menunjukkan diagram blok kontroler diferensial.



Gambar 2.13 Diagram Blok Kontroler Diferensial (Ogata K., 1997)

2.5.4 Kontroler Proporsional Integral Diferensial (PID)

Gabungan aksi kontrol proporsional, integral, dan diferensial mempunyai keunggulan dapat saling menutupi kekurangan dan kelebihan dari masing-masing kontroler. Persamaan kontroler PID ini dapat dinyatakan sebagai berikut :

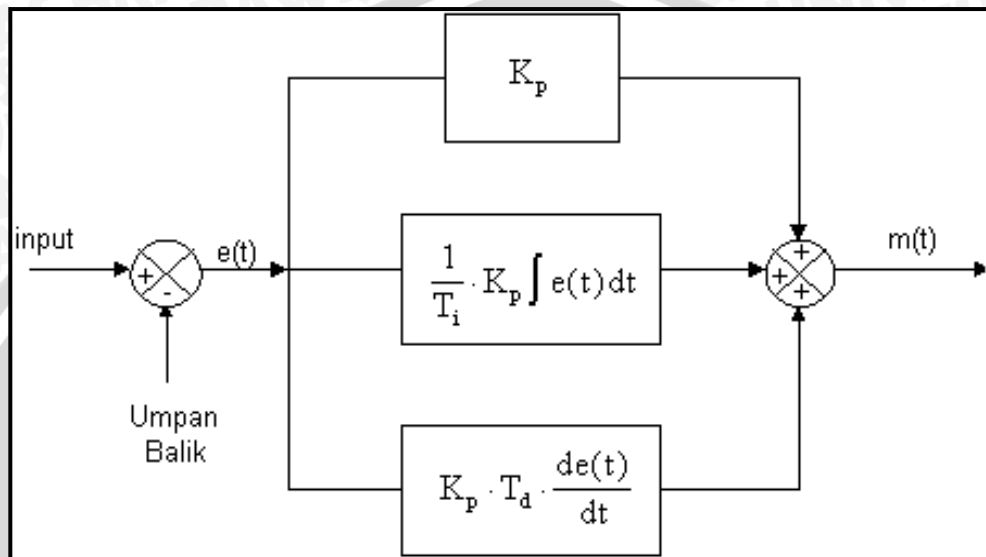
$$m(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \cdot e(t)dt + K_p \cdot T_d \frac{de(t)}{dt} \dots\dots\dots (2.14)$$

Dalam transformasi Laplace dinyatakan sebagai berikut :

$$\frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \dots\dots\dots (2.15)$$

dengan T_i adalah waktu integral, dan T_d adalah waktu derivatif.

Gambar 2.14 menunjukkan diagram blok kontroler PID.



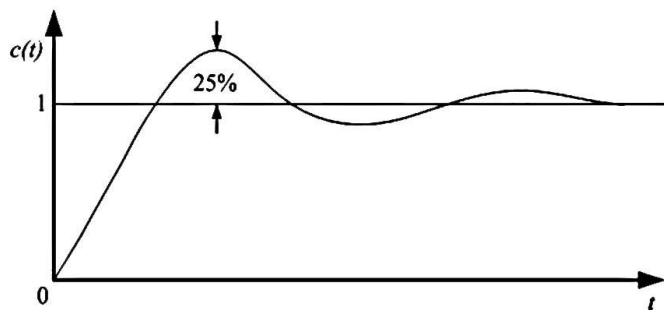
Gambar 2.14 Diagram Blok Kontroler PID (Ogata K., 1997)

2.5.5 Metode Perancangan Kontroler Proporsional Integral Diferensial (PID) Menggunakan Metode Ziegler-Nichols

Ziegler dan Nichols mengemukakan aturan-aturan untuk menentukan nilai dari gain proporsional K_p , waktu integral T_i , dan waktu derivatif T_d berdasarkan karakteristik respon transien dari *plant* yang diberikan. Penentuan parameter kontroler PID atau penalaan kontroler PID tersebut dapat dilakukan dengan bereksperimen dengan plan (Ogata K., 1997).

Terdapat dua metode yang disebut dengan aturan penalaan Ziegler-Nichols, pada kedua metode tersebut memiliki tujuan yang sama yaitu untuk mencapai 25% *maximum overshoot* pada respon unit step.

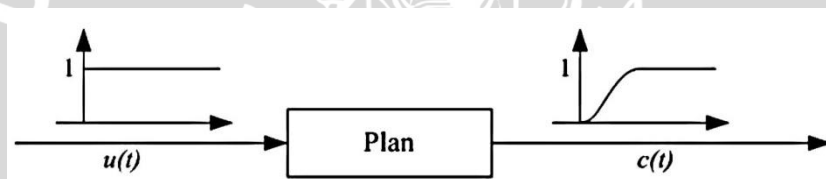
Kurva respon unit step yang menunjukkan 25% *maximum overshoot* terlihat dalam Gambar 2.15.



Gambar 2.15 Kurva Respon Unit Step Menunjukkan 25% *Maximum Overshoot* (Ogata K., 1997)

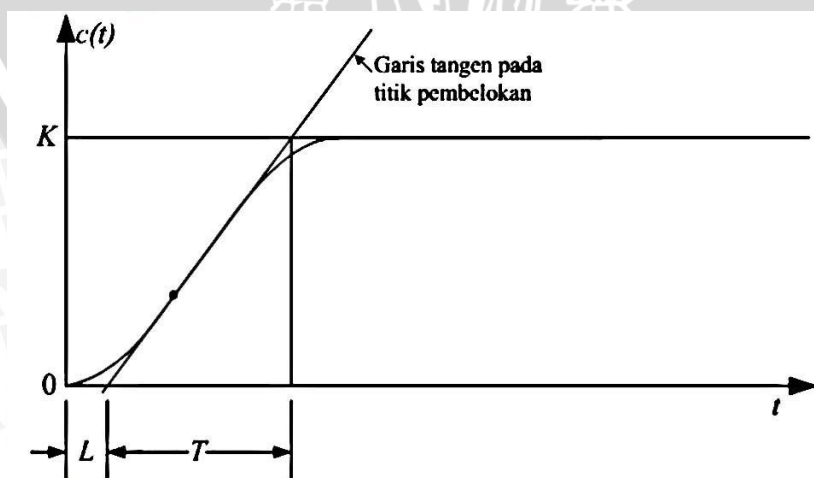
a) Metode Pertama

Metode pertama atau sering disebut metode kurva reaksi, respon dari plan dapat dapat diperoleh secara eksperimental dengan masukan berupa unit step, seperti yang ditunjukkan dalam Gambar 2.16.



Gambar 2.16 Respon Plant Terhadap Masukan Berupa Unit Step (Ogata K., 1997)

Jika dalam plan tersebut terdapat integrator atau *dominan complex-conjugate poles*, maka kurva respon unit step berbentuk seperti huruf S, seperti dalam Gambar 2.17, jika respon tidak memberikan bentuk kurva S, maka metode ini tidak berlaku.(Ogata K., 1997).



Gambar 2.17 Kurva yang Berbentuk S (Ogata K., 1997)

Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda L dan konstanta waktu T . Waktu tunda dan konstanta waktu ditentukan dengan menggambar sebuah garis tangen pada titik pembelokan dari kurva S, dan menentukan perpotongan antara garis tangen dengan sumbu waktu t dan sumbu $c(t) = K$, seperti yang telah ditunjukkan dalam Gambar 2.17. Fungsi alih $C(s)/U(s)$ dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut:

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts+1} \dots\dots\dots (2.16)$$

Ziegler dan Nichols menyarankan untuk menentukan nilai-nilai dari K_p , T_i dan T_d berdasarkan pada formula yang ditunjukkan dalam Tabel 2.2 (Ogata K., 1997).

Tabel 2.2 Aturan Penalaran Ziegler-Nichols Berdasarkan Respon Unit Step dari Plan (Ogata K., 1997)

Tipe Kontroler	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	$2L$	$0.5L$

Aturan untuk metode pertama dengan persamaan sebagai berikut:

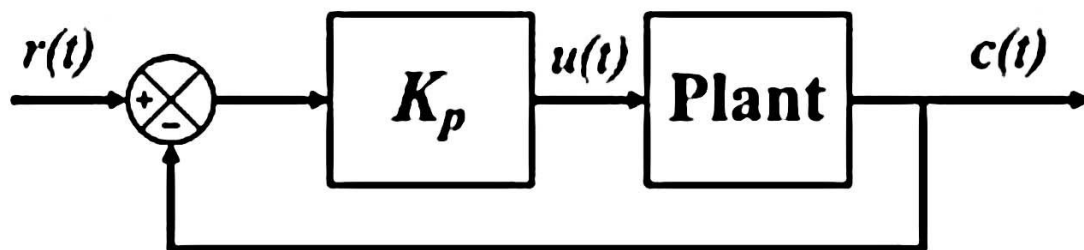
$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = 1.2 \frac{T}{L} \left(1 + \frac{1}{2Ls} + 0.5Ls \right) \dots\dots\dots (2.17)$$

$G_c(s)$ merupakan penguatan sistem yang memiliki nilai gain proporsional K_p , waktu integral T_i , dan waktu derivatif T_d .

b) Metode Kedua

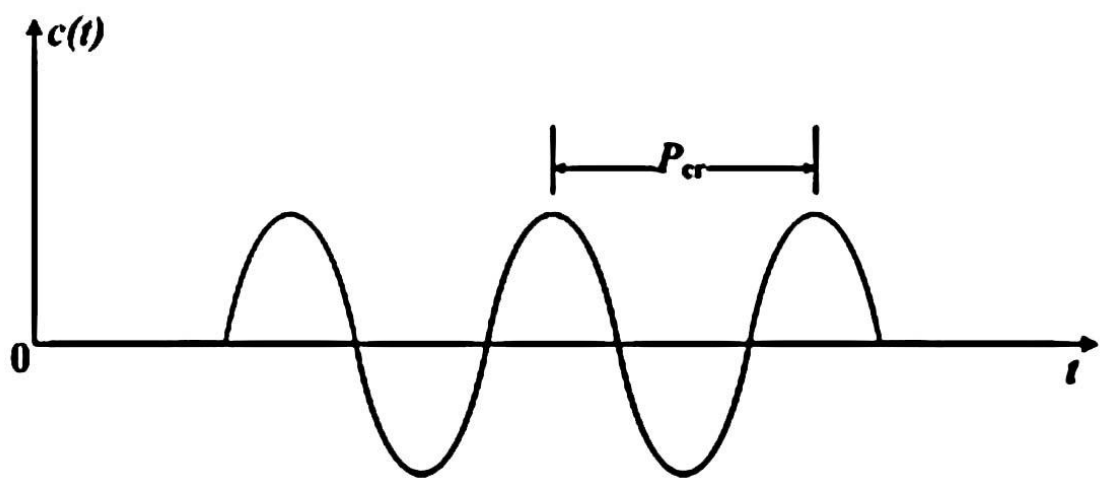
Dalam metode kedua Ziegler-Nichols, mula-mula yang dilakukan adalah membuat $T_i = \infty$ dan $T_d = 0$. Kemudian hanya dengan menggunakan tindakan kontrol proporsional, harga ditingkatkan dari nol ke suatu nilai kritis K_{cr} , disini mula-mula keluaran memiliki osilasi yang berkesinambungan (Jika keluaran tidak memiliki osilasi berkesinambungan untuk nilai K_p manapun yang telah diambil, maka metode ini tidak berlaku). Dari keluaran yang berosilasi secara berkesinambungan, penguatan kritis K_{cr} dan periode P_{cr} dapat ditentukan.

Diagram blok sistem loop tertutup dengan kontroler proporsional dapat dilihat dalam Gambar 2.18.



Gambar 2.18 Sistem Loop Tertutup dengan Kontroler Proporsional (Ogata K., 1997)

Osilasi berkesinambungan dengan periode P_{cr} dapat dilihat dalam Gambar 2.19.



Gambar 2.19 Osilasi Berkesinambungan dengan Periode P_{cr} (Ogata K., 1997)

Ziegler dan Nichols menyarankan penyetelan nilai parameter K_p , T_i , dan T_d berdasarkan rumus yang diperlihatkan dalam Tabel 2.3 (Ogata K., 1997).

Tabel 2.3 Aturan Dasar Ziegler-Nichols Berdasarkan *Critical Gain* K_{cr} dan *Critical Period* P_{cr} (Ogata K., 1997)

Tipe Kontroler	K_p	T_i	T_d
P	$0.5 K_{cr}$	∞	0
PI	$0.45 K_{cr}$	$\frac{1}{1.2} P_{cr}$	0
PID	$0.6 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

Aturan untuk metode kedua dengan persamaan sebagai berikut:

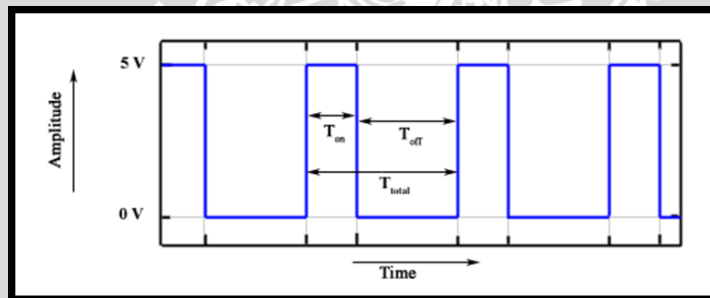
$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = 0.6K_{cr} \left(1 + \frac{1}{0.5P_{cr}s} + 0.125P_{cr}s \right) \dots\dots\dots (2.18)$$

Gc(s) merupakan penguatan sistem yang memiliki nilai gain proporsional Kp, waktu integral Ti, dan waktu derivatif Td.

2.6 Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) digunakan untuk mengatur kecepatan dari motor DC. Kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut. Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0% - 100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan.

Sinyal *Pulse Width Modulation* (PWM) secara umum dapat dilihat dalam Gambar 2.20 berikut:



Gambar 2.20 Sinyal PWM Secara Umum (insansains.wordpress.com, 2008)

$$Duty\ cycle = \frac{T_{on}}{T} \times 100\% (\%) \dots\dots\dots (2.19)$$

Dengan:

Ton = Periode logika tinggi

T = Periode keseluruhan

$$V_{dc} = Duty\ cycle \times V_{cc} (V) \dots\dots\dots (2.20)$$

Sedangkan frekuensi sinyal dapat ditentukan dengan rumus berikut:

$$f_{On} = \frac{f_{clk\ 1/0}}{N.256} (Hz) \dots\dots\dots (2.21)$$

2.7 Mikrokontroler

Mikrokontroler populer yang pertama dibuat oleh Intel pada tahun 1976, yaitu mikrokontroler 8-bit Intel 8748. Mikrokontroler tersebut adalah bagian dari keluarga mikrokontroler MCS-48. Sebelumnya, Texas *instruments* telah memasarkan mikrokontroler 4-bit pertama yaitu TMS 1000 pada tahun 1974. TMS 1000 yang mulai dibuat sejak 1971 adalah mikrokomputer dalam sebuah *chip*, lengkap dengan RAM dan ROM.

Pengendali mikro (Inggris: *microcontroller*) adalah sistem mikroprosesor lengkap yang terkandung di dalam sebuah *chip*. Mikrokontroler berbeda dari mikroprosesor serba guna yang digunakan dalam sebuah PC, karena sebuah mikrokontroler umumnya telah berisi komponen pendukung sistem minimal mikroprosesor, yakni memori dan antarmuka I/O.

Berbeda dengan CPU serba-guna, mikrokontroler tidak selalu memerlukan memori eksternal, sehingga mikrokontroler dapat dibuat lebih murah dalam kemasan yang lebih kecil dengan jumlah *pin* yang lebih sedikit.

Sebuah *chip* mikrokontroler umumnya memiliki fitur:

- a) *central processing unit* - mulai dari prosesor 4-bit yang sederhana hingga prosesor kinerja tinggi 64-bit.
- b) *input/output* antarmuka jaringan seperti *port serial* (UART)
- c) antarmuka komunikasi serial lain seperti *PC, Serial Peripheral Interface* and *Controller Area Network* untuk sambungan sistem
- d) periferal seperti *timer* dan *watchdog*
- e) RAM untuk penyimpanan data
- f) ROM, EPROM, EEPROM atau *Flash memory* untuk menyimpan program komputer
- g) pembangkit *clock* - biasanya berupa resonator rangkaian RC
- h) pengubah analog-ke-digital

2.8 Mikrokontroler Atmel AVR ATmega 8535

ATmega 8535 merupakan salah satu mikrokontroler 8 bit buatan Atmel berbasis arsitektur *Reduced Instruction Set Computer* (RISC).

ATmega 8535 memiliki beberapa kemamuan yaitu:

1. Sistem mikrokontroler 8 bit berbasis RISC dengan kecepatan maksimal 16 MHz.
2. Memiliki memori flash 8 kB, SRAM sebesar 512 byte dan *Electrically Erasable Programmable Read Only Memory* (EEPROM) sebesar 512 byte.
3. Memiliki *Analog-to-Digital Converter* (ADC) internal dengan ketelitian 10 bit sebanyak 8 saluran.
4. Memiliki *Pulse Width Modulation* (PWM) internal sebanyak 4 saluran.
5. Portal komunikasi serial (USART) dengan kecepatan maksimal 2,5 Mbps.
6. Enam pilihan mode *sleep*, untuk menghemat penggunaan daya listrik.

Mikrokontroler ATmega 8535 memiliki 40 pin untuk model PDIP dengan nama-nama tiap pin yaitu:

1. VCC untuk tegangan pencatu daya positif.
2. GND untuk tegangan pencatu daya negative.
3. PortA (PA0-PA7) sebagai port *Input/Output* dan memiliki kemampuan lain yaitu sebagai *input* untuk ADC.
4. PortB (PB0-PB7) sebagai port *Input/Output* dan memiliki kemampuan yang lain.
5. PortC (PC0-PC7) sebagai port *Input/Output* untuk ATmega8535.
6. PortD (PD0-PD7) sebagai port *Input/Output* dan memiliki kemampuan yang lain.
7. RESET untuk melakukan reset program dalam mikrokontroler.
8. XTAL1 dan XTAL2 untuk *input* pembangkit sinyal *clock*.
9. AVCC untuk pin masukan tegangan pencatu daya untuk ADC.
10. AREF untuk pin tegangan referensi ADC.

Konfigurasi pin ATmega 8535 dapat dilihat dalam Gambar 2.21.

PDIP			
(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(SS) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5
(TXD) PD1	15	26	PC4
(INT0) PD2	16	25	PC3
(INT1) PD3	17	24	PC2
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)

Gambar 2.21 Konfigurasi Pin ATmega 8535 PDIP (ATMEL, 2010:2)

2.9 Program CodeVision AVR

CodeVisionAVR merupakan sebuah cross-compiler C, *Integrated Development Environment* (IDE), dan *Automatic Program Generator* yang didesain untuk mikrokontroler buatan Atmel seri AVR. CodeVisionAVR dapat dijalankan pada sistem operasi Windows 95, 98, Me, NT4, 2000, dan XP.

Cross-compiler C mampu menerjemahkan hampir semua perintah dari bahasa ANSI C, sejauh yang diijinkan oleh arsitektur dari AVR, dengan tambahan beberapa fitur untuk mengambil kelebihan khusus dari arsitektur AVR dan kebutuhan pada sistem *embedded*.

File object COFF hasil kompilasi dapat digunakan untuk keperluan *debugging* pada tingkatan C, dengan pengamatan variabel, menggunakan *debugger* Atmel AVR Studio.

IDE mempunyai fasilitas internal berupa *software AVR Chip In-System Programmer* yang memungkinkan pengguna untuk melakukan transfer program kedalam chip mikrokontroler setelah sukses melakukan kompilasi/assembly secara otomatis. *Software In-System Programmer* didesain untuk bekerja dengan Atmel STK500/AVRISP/AVRProg, Kanda Systems STK200+/300, Dontronics DT006, Vogel

Elektronik VTEC-ISP, Futurlec JRAVR dan MicroTronics ATCPU/Mega2000 programmers/development boards.

Untuk keperluan *debugging* sistem *embedded*, yang menggunakan komunikasi serial, IDE mempunyai fasilitas internal berupa sebuah Terminal.

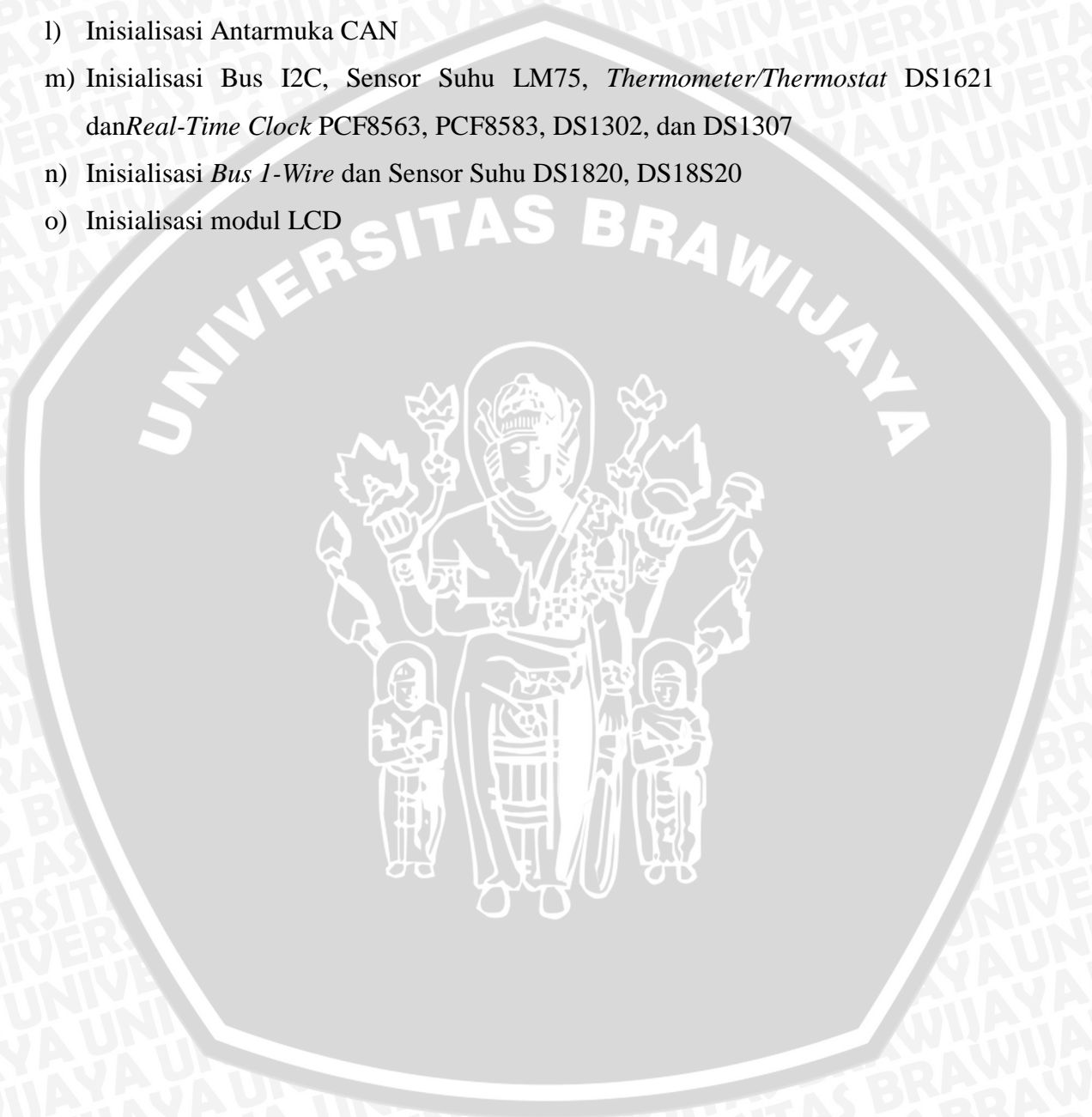
Selain *library* standar C, CodeVisionAVR juga mempunyai *library* tertentu untuk:

- a) Modul LCD alphanumeric
- b) Bus I2C dari Philips
- c) Sensor Suhu LM75 dari National Semiconductor
- d) *Real-Time Clock*: PCF8563, PCF8583 dari Philips, DS1302 dan DS1307 dari Maxim/Dallas Semiconductor
- e) Protokol 1-Wire dari Maxim/Dallas Semiconductor
- f) Sensor Suhu DS1820, DS18S20, dan DS18B20 dari Maxim/Dallas Semiconductor
- g) Termometer/Termostat DS1621 dari Maxim/Dallas Semiconductor
- h) EEPROM DS2430 dan DS2433 dari Maxim/Dallas Semiconductor
- i) SPI
- j) *Power Management*
- k) Delay
- l) Konversi ke Kode Gray

CodeVisionAVR juga mempunyai Automatic Program Generator bernama CodeWizardAVR, yang mengizinkan pengguna untuk menulis, dalam hitungan menit, semua instruksi yang diperlukan untuk membuat fungsi-fungsi berikut:

- a) *Set-up* akses memori eksternal
- b) Identifikasi sumber *reset* untuk *chip*
- c) Inisialisasi port *input/output*
- d) Inisialisasi interupsi eksternal
- e) Inisialisasi *Timer/Counter*
- f) Inisialisasi *Watchdog-Timer*
- g) Inisialisasi UART (USART) dan komunikasi *serial* berbasis *buffer* yang digerakkan oleh interupsi

- h) Inisialisasi Pembanding Analog
- i) Inisialisasi ADC
- j) Inisialisasi Antarmuka SPI
- k) Inisialisasi Antarmuka *Two-Wire*
- l) Inisialisasi Antarmuka CAN
- m) Inisialisasi Bus I2C, Sensor Suhu LM75, *Thermometer/Thermostat* DS1621 dan *Real-Time Clock* PCF8563, PCF8583, DS1302, dan DS1307
- n) Inisialisasi *Bus 1-Wire* dan Sensor Suhu DS1820, DS18S20
- o) Inisialisasi modul LCD



BAB III

METODOLOGI PENELITIAN

Kajian dalam skripsi ini merupakan penelitian yang bersifat aplikatif, yaitu merancang dan membuat suatu *Hovercraft Line Follower* dengan menitikberatkan pengendalian kecepatan pada motor pendorong dengan menggunakan kontroler Proporsional Integral Diferensial (PID) yang bertujuan agar dapat menampilkan performansi sistem sesuai dengan yang direncanakan.

Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat adalah sebagai berikut:

1. Spesifikasi alat
2. Perancangan dan realisasi pembuatan alat
3. Pengujian alat
4. Pengambilan kesimpulan

3.1 Spesifikasi Alat

Adapun spesifikasi alat yang akan direalisasikan adalah sebagai berikut:

1. *Hovercraft* yang digunakan jenis *Hovercraft Flexible Skirt*.
2. *Hovercraft* memiliki dimensi lebar 28 cm, panjang 45 cm, dan tinggi 24 cm. Menggunakan *acrylic* 2 mm dan 3 mm sebagai kerangka serta *Styrofoam* sebagai sekat penghubung rangka.
3. Mikrokontroler yang digunakan ialah ATMEGA 8535 keluaran Atmel.
4. Kontroler yang digunakan ialah kontroler Proporsional, Deferensial dan Integral (PID).
5. Pemrograman menggunakan bahasa pemrograman tingkat ke tiga, yaitu C++.
6. Rangkaian mikrokontroler, sensor, dan *electronic speed controller* (ESC) terletak dalam lambung sedangkan tombol *tuning* dan LCD terletak pada bagian depan.
7. Motor DC *brushless* 1400kV terletak pada lambung mengarah ke dalam dan pada bagian belakang sebagai pendorong dengan masing-masing menggunakan *propeller* 6 x 5 inci.

3.2 Perancangan dan Realisasi Pembuatan Alat

3.2.1 Perancangan Perangkat Keras dan Realisasi Pembuatan Alat

- Pembuatan diagram blok secara lengkap
- Penentuan dan perhitungan komponen yang akan digunakan
- Merakit perangkat keras (*hardware*) untuk masing-masing blok.

3.2.2 Perancangan dan Perhitungan Komponen yang akan Digunakan

Setelah mengetahui seperti apa perangkat keras yang dirancang, maka dibutuhkan perangkat lunak untuk mengendalikan dan mengatur kerja dari alat ini. Desain dan parameter yang telah dirancang, kemudian ditetapkan pada mikrokontroler dengan menggunakan bahasa pemrograman C++.

3.2.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan setelah mengetahui nilai parameter Proporsional (P), Integral (I), dan Diferensial (D). Perancangan dimulai dari pembuatan *flowchart*, kemudian dilanjutkan dengan penulisan *listing code*.

3.3 Pengujian Alat

Setelah semua komponen pada alat sudah terhubung sesuai dengan diagram blok sistem yang telah dirancang dan perangkat lunak untuk mendukung sistem yang telah dibuat, maka diadakan pengujian dan analisa alat. Metode pengujiannya adalah sebagai berikut:

1. Pengujian Catu Tegangan Baterai

Pengujian baterai dilakukan dengan cara menghubungkan dengan multimeter pengukur tegangan. Pengujian ini bertujuan untuk mengetahui tegangan yang dihasilkan oleh baterai. Baterai yang akan diuji adalah baterai *li-po* 11.1 V 2200 mA.

2. Pengujian Tegangan Regulator dan ESC

Pengujian dilakukan dengan cara memberi catu rangkaian regulator +5 V dan ESC dengan baterai *li-po* dan diukur tegangan keluarannya. Pengujian ini bertujuan untuk membandingkan hasil tegangan keluaran yang akan digunakan untuk memberi catu mikrokontroler. Hasil keluaran dipilih tegangan yang stabil mendekati +5 V.

3. Pengujian Sensor

Pengujian sensor dilakukan dengan cara mensimulasikan rangkaian sensor dan hasil pemodelan rangkaian sensor. Pengujian ini bertujuan untuk memastikan sensor dan hasil pemodelan sensor dapat bekerja sesuai dengan perancangan dan memberikan analisa terhadap hasil pengujian. Terdapat rangkaian sensor utama yang akan diuji, yaitu sensor *photo reflective* sebagai pengukur intensitas cahaya yang dikonversi ke dalam tegangan.

4. Pengujian RPM dan *Thrush* Motor DC *Brushless Hover*

Pengujian ini dilakukan dengan mensimulasikan berbagai macam kecepatan putaran motor yang telah dipasang *propeller* untuk melihat RPM dan gaya angkat yang dihasilkan. Pengujian ini bertujuan untuk membandingkan *thrust* secara perhitungan dan pengujian.

5. Pengujian Sinyal PWM Motor Servo

Pengujian ini dilakukan dengan mensimulasikan macam sudut arah servo yang telah dipasang sirip untuk melihat sudut terbaik yang dapat mengarahkan gerak robot dalam mengikuti garis hitam.

6. Pengujian Massa Robot *Hovercraft Line Follower*

Pengujian ini dilakukan dengan cara menimbang keseluruhan robot untuk mengetahui massa rata-rata yang digunakan sebagai nilai ukur untuk menghitung gaya berat.

7. Pengujian Nilai Kritis Kcr

Pengujian ini dilakukan dengan cara memberi nilai kenaikan kontrol proporsional secara bertahap. Pengujian ini bertujuan untuk mendapatkan nilai kritis Kcr dan respon sistem osilasi yang berkelanjutan.

8. Pengujian RPM Motor DC *Brushless* Pendorong

Pengujian ini dilakukan dengan cara mensimulasikan robot saat mengikuti garis dengan dan tanpa kontroler. Pengujian ini bertujuan untuk mengetahui apakah keluaran kontroler sudah sesuai dengan yang direncanakan atau tidak.

9. Pengujian Sistem Keseluruhan

Pengujian ini dilakukan dengan cara menggabungkan semua bagian alat yang dibuat dan melihat kinerja alat. Pengujian ini bertujuan untuk mengetahui kinerja alat yang dibuat dan memberikan analisa terhadap kinerja alat.

3.4 Pengambilan Kesimpulan

Kesimpulan diambil berdasarkan data yang diperoleh dari pengujian sistem secara keseluruhan. Jika hasil yang didapatkan telah sesuai dengan yang direncanakan sebelumnya, maka sistem kendali tersebut telah berhasil memenuhi harapan dan tentunya memerlukan pengembangan lebih lanjut untuk penyempurnaan.



BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

Perancangan dan pembuatan alat ini terdiri dari dua bagian, yaitu bagian perangkat keras dan bagian perangkat lunak. Perancangan dan pembuatan alat dilakukan secara bertahap untuk memudahkan analisa sistem. Beberapa aspek yang perlu dijelaskan dalam bab ini meliputi penentuan spesifikasi alat, perencanaan masing-masing blok rangkaian dan perencanaan sistem secara keseluruhan.

4.1 Spesifikasi Sistem

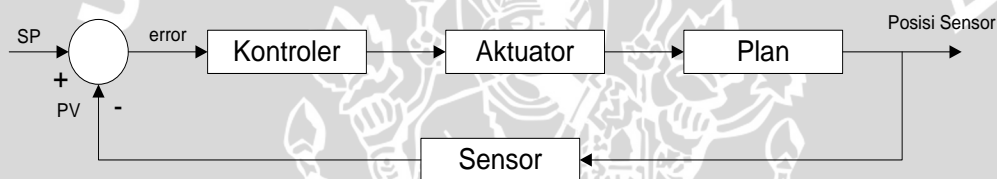
Spesifikasi sistem yang dirancang adalah sebagai berikut:

1. Robot *Hovercraft Line Follower* menggunakan rangka dengan spesifikasi sebagai berikut:
 - *Body* atas dan bawah memiliki ukuran panjang 35 cm dan lebar 20 cm serta dibentuk sesuai desain.
 - Penyangga sirip dengan panjang 17 cm, lebar 2 cm, dan tinggi 17 cm serta 3 sirip dengan panjang 16 cm dan lebar 10 cm .
 - *Body*, penyangga sirip dan sirip menggunakan bahan *acrylic* dengan tebal 2 mm.
 - Penyangga motor *hover* dengan panjang 20 cm dan lebar 4.4 cm serta penyangga motor pendorong dengan panjang 5 cm, lebar 4.4 cm, dan tinggi 12 cm.
 - Penyangga motor *hover* dan motor pendorong menggunakan bahan *acrylic* dengan tebal 3 mm.
 - Sekat untuk menggabungkan *body* atas dan bawah menggunakan *Styrofoam* dengan tebal 1.5cm.
 - *Skirt* untuk perangkap udara menggunakan plastik hitam dengan panjang 49 cm dan lebar 34 cm serta dibentuk sesuai desain.
2. Pergerakan robot *Hovercraft Line Follower* menggunakan 2 (dua) buah motor DC *brushless* dengan 1 (satu) buah motor untuk menghasilkan gaya angkat dan 1 (satu) buah motor untuk menghasilkan gaya dorong.

3. *kV rating* pada motor DC *brushless* adalah 1400kV dengan maksimum tegangan masukan adalah 11.1 V.
4. *Propeller* pada *shaft* motor menggunakan EMP *Electric Propeller* 6x5E (diameter 6 inci).
5. 2 (dua) *Electronic Speed Controller* sebagai pengkondisi sinyal masukan pada motor DC *brushless* sebesar 30 A.
6. Menggunakan 8 (delapan) buah sensor *photo reflective*, enam berada di bagian depan dan satu di pojok belakang samping kanan dan kiri.
7. Mikrokontroler yang digunakan adalah ATMEGA 8535.
8. Catu tegangan menggunakan baterai *Li-po* 11.1 V 2200mA.

4.2 Diagram Blok Sistem

Diagram blok secara umum ditunjukkan dalam Gambar 4.1.

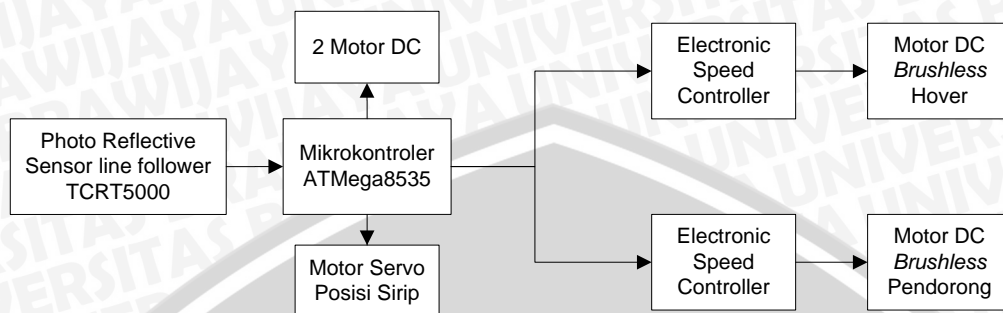


Gambar 4.1 Diagram Blok Sistem secara Umum (Perancangan, 2013)

Keterangan diagram blok dalam Gambar 4.1 adalah:

- *Input* berupa nilai parameter dari pembacaan sensor saat posisi lurus *setpoint* (SP) = 0.
- Sinyal *feedback* dari sensor berupa bobot sensor (PV) pembacaan sensor yang diumpanbalikkan.
- *Error* berupa simpangan antar bobot sensor (PV) dengan *setpoint* (SP).
- Kontroler yang digunakan adalah Proporsional, Integral, dan Diferensial dengan menggunakan mikrokontroler untuk mengolah nilai PID. Nilai PID digunakan sebagai nilai referensi pemanggilan nilai kontrol kecepatan motor DC *brushless*.
- Aktuator berupa 2 (dua) motor DC *brushless*, satu motor servo, dan 2 (dua) motor DC magnet permanen.
- *Plan* berupa robot *hovercraft line follower*.
- *Output* berupa nilai posisi sensor yang dipertahankan pada nilai 0 dengan mengendalikan kecepatan motor pendorong.

Sistem pergerakan robot *hovercraft line follower* secara keseluruhan ditunjukkan dalam Gambar 4.2.



Gambar 4.2 Diagram Blok Sistem Robot *Hovercraft Line Follower* (Perancangan, 2013)

Keterangan diagram blok dalam Gambar 4.2 adalah:

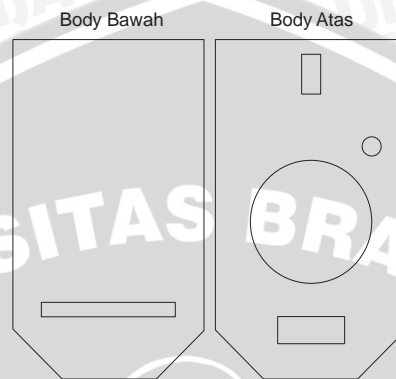
- *Photo Reflective* memberikan nilai $SP = 0$ dan nilai PV sesuai nilai bobot pembacaan yang diberikan tiap sensor.
- Mikrokontroler mengolah nilai *error* dari simpangan antara nilai SP sebagai *input* dan PV untuk menghasilkan sinyal kontrol yang kemudian dikirimkan ke motor servo, 2 motor DC, dan *electronic speed controller* (ESC).
- Mikrokontroler memberikan sinyal kontrol menuju ESC motor DC *brushless hover*, mengolah sinyal dan menghasilkan sinyal kontrol PID sebagai referensi nilai pengurangan PWM yang kemudian dikirimkan menuju ESC motor DC *brushless* pendorong.
- ESC berfungsi menguatkan sinyal kontrol dan diteruskan menuju motor DC *brushless*.
- Keluaran akan mempengaruhi posisi robot sehingga akan memberikan nilai bobot sensor (PV) pembacaan sensor baru.
- Hasil pembacaan sensor kemudian dikurangkan dengan *input* sehingga mikrokontroler mampu mengkompensasi *error* yang terjadi.

4.3 Perancangan Perangkat Keras

Perangkat keras terdiri dari rangka *hovercraft line follower*, board mikrokontroler, board sensor *line tracer*, *electronic speed controller*, motor DC *brushless*, dan *propeller* dan sistem kendali arah.

4.3.1 Perancangan Rangka Robot *Hovercraft Line Follower*

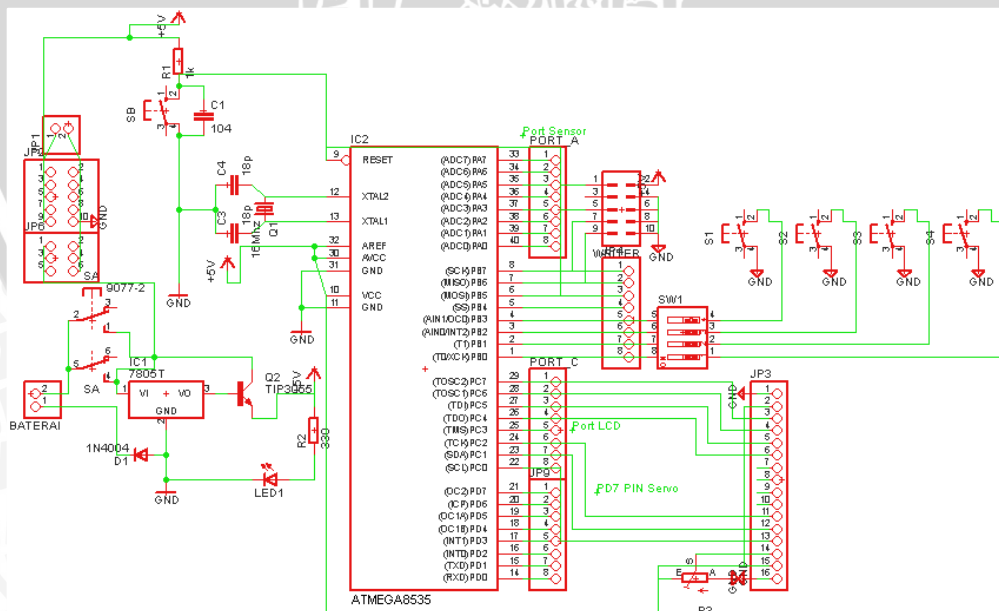
perancangan rangka dilakukan sebagai dasar dari komponen-komponen yang akan diletakkan. Secara umum rangka robot *hovercraft line follower* ditunjukkan dalam Gambar 4.3.



Gambar 4.3 Rangka Robot *Hovercraft Line Follower* (Perancangan, 2013)

4.3.2 Rangkaian *Board* Mikrokontroler

Board mikrokontroler merupakan rangkaian utama dari sistem robot *hovercraft line follower*. Mikrokontroler terhubung langsung dengan sensor *photo reflective*, motor servo dan dua buah ESC. Pada pengendalian robot *hovercraft line follower* digunakan ATMEGA 8535. Rangkaian *board* mikrokontroler ditunjukkan dalam Gambar 4.4:



Gambar 4.4 Rangkaian *Board* Mikrokontroler (Perancangan, 2013)

Mikrokontroler 8535 memiliki 32 jalur yang dapat diprogram menjadi keluaran atau masukan. *Pin* masukan dan keluaran mikrokontroler pada perancangan akan difungsikan sesuai Tabel 4.1:

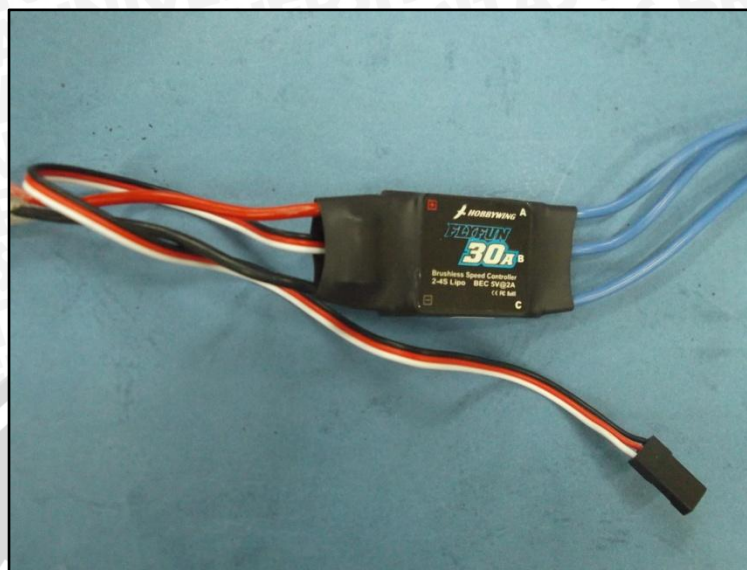
Tabel 4.1 Fungsi *Pin* Mikrokontroler (Perancangan, 2013)

No.	<i>Pin</i>	Fungsi
1	PA0	Jalur masukan sensor 0
2	PA1	Jalur masukan sensor 1
3	PA2	Jalur masukan sensor 2
4	PA3	Jalur masukan sensor 3
5	PA4	Jalur masukan sensor 4
6	PA5	Jalur masukan sensor 5
7	PA6	Jalur masukan sensor 6
8	PA7	Jalur masukan sensor 7
9	PB0	Jalur masukan <i>tuning cancel</i>
10	PB1	Jalur masukan <i>tuning up</i>
11	PB2	Jalur masukan <i>tuning down</i>
12	PB3	Jalur masukan <i>tuning ok</i>
13	PB4	Jalur keluaran hubungan SPI
14	PB5	Jalur keluaran hubungan SPI
15	PB6	Jalur keluaran hubungan SPI
16	PB7	Jalur keluaran hubungan SPI
17	PC0	Jalur keluaran <i>Register Select</i> (LCD)
18	PC1	Jalur keluaran <i>Read/ Write</i> (LCD)
19	PC2	Jalur keluaran <i>Enable</i> (LCD)
20	PC4	Jalur keluaran DB4 (LCD)
21	PC5	Jalur keluaran DB5 (LCD)
22	PC6	Jalur keluaran DB6 (LCD)
23	PC7	Jalur keluaran DB7 (LCD)
24	PD7	Jalur keluaran motor servo

4.3.3 *Electronic Speed Controller* (ESC)

Electronic speed controller berfungsi sebagai penguat tegangan dari keluaran *pin* motor dan juga berfungsi sebagai pemecah fasa. Karena motor DC *brushless* yang digunakan merupakan motor DC dengan 3 fasa dengan hubungan *star*.

ESC yang digunakan ditunjukkan dalam Gambar 4.5.



Gambar 4.5 *Electronic Speed Controller 30A* (Perancangan, 2013)

4.3.4 Pemilihan Motor DC *Brushless* dan *Propeller*

Kombinasi antara motor DC *brushless* dan *propeller* harus tepat agar mampu menghasilkan gaya angkat (*thrust*) yang sesuai dengan berat keseluruhan robot *hovercraft line follower* (termasuk *battery*), motor DC *brushless* dan komponen-komponen lainnya. Perhitungan mengenai *thrust* yang dihasilkan sebuah *propeller* didasarkan pada persamaan 4.1 berikut ini:

$$T = cT \frac{4\rho r^4}{\pi^2} \omega^2 \dots\dots\dots (4.1)$$

- dimana:
- T = *thrust propeller* (N)
 - cT = koefisien *propeller* (0.1098 untuk EMP 6x5E)
 - ρ = kerapatan udara (1.225 kg/m³)
 - ω = kecepatan anguler *propeller* (2πn satuan rad/s)
 - n = jumlah putaran *propeller* per detik
 - r = jari-jari *propeller* (m)
 - π = 3.1429

Sesuai dengan spesifikasi sistem yang dijabarkan sebelumnya, bahwa pada skripsi ini menggunakan *electric model power* (EMP) 6x5E dengan motor DC *brushless* 1400kV. EMP 6x5E memiliki diameter 15.24 cm yang memadai untuk diposisikan di dalam lambung robot yang memiliki lebar 20 cm.

Tabel 4.2 menunjukkan perhitungan *thrust* yang dihasilkan setiap kenaikan kecepatan motor DC *brushless*.

Tabel 4.2 Perhitungan *Thrust* (Perancangan, 2013)

RPM	ω (rad/s)	Thrust (N)
700	73.334	0.010
1400	146.669	0.040
2100	220.003	0.089
2800	293.337	0.158
3500	366.672	0.247
4200	440.006	0.356
4900	513.340	0.484
5600	586.675	0.632
6300	660.009	0.800
7000	733.343	0.988
7700	806.678	1.195
8400	880.012	1.422
9100	953.346	1.669
9800	1026.681	1.936
10500	1100.015	2.222
11200	1173.349	2.528
11900	1246.684	2.854
12600	1320.018	3.200
13300	1393.352	3.565
14000	1466.687	3.950

Jika berat maksimal *hovercraft line follower* mencapai 1.5 kg, maka perhitungan gaya berat *hovercraft line follower* adalah:

$$F = ma \dots\dots\dots (4.2)$$

- dengan
- F = gaya berat (N)
 - m = massa benda (kg)
 - a = gravitasi bumi (9.81 m/s²)

maka :

$$F = 1.5 \text{ kg} \times 9.81 \text{ m/s}^2$$

$$= 14.715 \text{ N}$$

Kemudian dibandingkan dengan tekanan dari gaya *thrust* dengan persamaan sebagai berikut:

$$p = \frac{F}{A} \dots\dots\dots (4.3)$$

- dengan
- p = tekanan (N/m²)
 - F = gaya berat (N)
 - A = luas penampang dasar robot (m²)

maka:

$$p_{robot} = \frac{14.715 \text{ N}}{0.0887 \text{ m}^2}$$

$$p_{robot} = 165.896 \text{ N/m}^2$$

$$p_{thrust} = \frac{2.854 \text{ N}}{0.0887 \text{ m}^2} \text{ (T pada RPM 11.900)}$$

$$p_{thrust} = 32.176 \text{ N/m}^2$$

Dari permisalan bahwa tekanan ke atas yang dibutuhkan pada *thrust* 2.854 N dengan 11.900 RPM. Maka akan terjadi tekanan dalam ruang udara sebesar 32.176 N/m².

Bentuk fisik *propeller* EMP 6x5E ditunjukkan dalam Gambar 4.6.



Gambar 4.6 *Propeller* EMP 6x5E (Perancangan, 2013)

Propeller EMP 6x5E dipilih untuk digunakan karena memiliki dimensi yang sesuai dengan dimensi robot, dan *propeller* EMP 6x5E yang termasuk *propeller* kecil juga sesuai dengan kemampuan putaran motor DC *brushless* C2822-1400kV.

Motor DC *brushless* 1400kV mampu menghasilkan putaran hingga 15540 rpm dengan tegangan masukan 11.1 V yang terlihat dalam Gambar 4.7.



Gambar 4.7 Motor DC *Brushless* 1400kV (Perancangan, 2013)

4.3.5 Pemilihan Sistem Kendali Arah

Kendali arah memanfaatkan tiga motor, yaitu satu motor servo dan dua motor DC. Motor servo sebagai penggerak sirip arah aliran hembusan udara BLDC pendorong harus memiliki torsi yang tinggi. Pada motor servo Toward Pro MG996R mampu menghasilkan torsi 9.4 kg/cm dalam tegangan 4.8 V. Alasan dipilihnya torsi tinggi supaya pergerakan sirip tidak terpengaruh oleh tekanan udara dari BLDC pendorong. Motor servo Toward Pro MG996R ditunjukkan dalam Gambar 4.8.

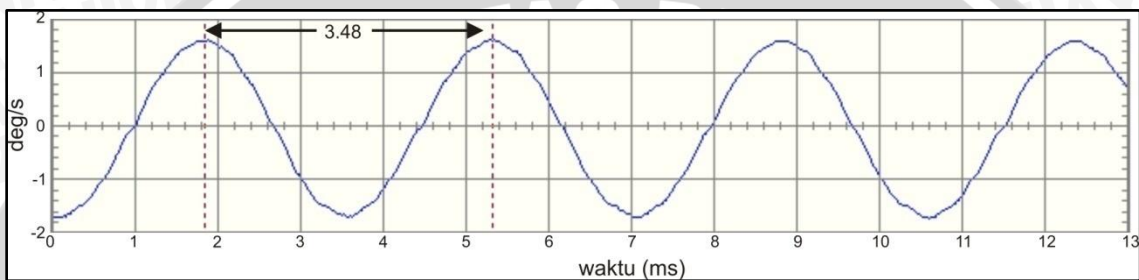


Gambar 4.8 Motor Servo Toward Pro MG996R (Perancangan, 2013)

Sedangkan dua motor DC magnet permanen dipilih yang memiliki putaran tinggi untuk membantu bagian depan robot berbelok tajam.

4.4 Penentuan Nilai Penguatan Kontroler

Perancangan kontroler dilakukan dengan menggunakan alat PC-SCOPE PCSU1000 1Gs/s – 2x60 MHz dengan *software* PCLab2000SE v4.04 untuk mengamati respon sistem. Langkah pertama adalah dengan melihat respon sistem yang osilasi setelah diberi kenaikan nilai $K_{cr} = 20$ seperti dalam Gambar 4.9:



Gambar 4.9 Respon Sistem (Perancangan, 2013)

Berdasarkan Gambar 4.9, didapatkan nilai P_{cr} sebesar 3.48. nilai parameter kontroler PID ditentukan berdasarkan tabel aturan dasar Ziegler-Nichols dengan *critical gain* K_{cr} dan *critical period* P_{cr} yang ditunjukkan dalam Tabel 4.3.

Tabel 4.3 Aturan Dasar Ziegler-Nichols Berdasarkan *Critical Gain* K_{cr} dan *Critical Period* P_{cr} (Ogata K., 1997)

Tipe Kontroler	K_p	T_i	T_d
P	$0.5 K_{cr}$	∞	0
PI	$0.45 K_{cr}$	$\frac{1}{1.2} P_{cr}$	0
PID	$0.6 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

maka nilai parameter PID yang diperoleh adalah:

$$K_p = 0.6 \times 20 = 12$$

$$T_i = 0.5 \times 3.48 = 1.74$$

$$T_d = 0.125 \times 3.48 = 0.435$$

dengan demikian dapat ditentukan nilai K_p , K_i , dan K_d adalah:

$$K_p = 12$$

$$K_i = \frac{K_p}{T_i}$$

$$= \frac{12}{1.74}$$

$$= 6.89655$$

$$= 6.897$$

$$Kd = Kp \times Td$$

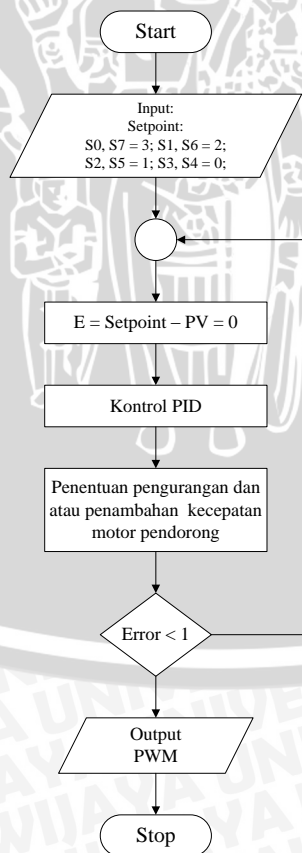
$$= 12 \times 0.435$$

$$= 5.22$$

Penentuan nilai penguatan dari metode kedua Zigler-Nichols diperoleh $Kp = 12$, $Ki = 6.897$, dan $Kd = 5.22$ yang akan digunakan untuk perhitungan nilai PID pada sistem.

4.5 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada skripsi ini menggunakan bahasa pemrograman C++ dengan menggunakan *software* CodeVision AVR. *Tuning* kontroler PID adalah dengan menggunakan metode *hand tuning*. *Flowchart* perancangan perangkat lunak ditunjukkan dalam Gambar 4.10.



Gambar 4.10 Flowchart Perangkat Lunak (Perancangan, 2013)

BAB V

PENGUJIAN DAN ANALISIS

Tujuan pengujian sistem ini adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan. Pengujian dibagi menjadi beberapa bagian, yaitu:

1. Pengujian catu tegangan baterai
2. Pengujian tegangan regulator dan ESC
3. Pengujian tegangan keluaran sensor *photo reflective*
4. Pengujian RPM dan *thrust* motor DC *brushless hover*
5. Pengujian sinyal PWM motor servo
6. Pengujian massa robot *hovercraft line follower*
7. Pengujian nilai kritis Kcr
8. Pengujian RPM motor DC *brushless* pendorong
9. Pengujian sistem keseluruhan

5.1 Pengujian Catu Tegangan Baterai

Pengujian ini bertujuan untuk mengetahui catu tegangan yang dihasilkan oleh baterai *Li-po* yang digunakan untuk mencatu sistem keseluruhan.

5.1.1 Peralatan Pengujian

1. Baterai *Li-po* 11.1 V 2200 mA
2. Digital Multimeter SANWA CD772

5.1.2 Prosedur Pengujian

Pengujian dilakukan dengan menghubungkan kutub positif dan kutub negatif dengan multimeter digital. Kemudian dicatat hasil pengujian dan diulang sebanyak 10 kali pengujian.

5.1.3 Hasil Pengujian

Setelah melakukan prosedur pengujian, maka didapatkan hasil yang ditunjukkan dalam Tabel 5.1:

Tabel 5.1 Hasil Pengujian Catu Tegangan Baterai *Li-po* 11.1 V 2200 mA (Pengujian, 2013)

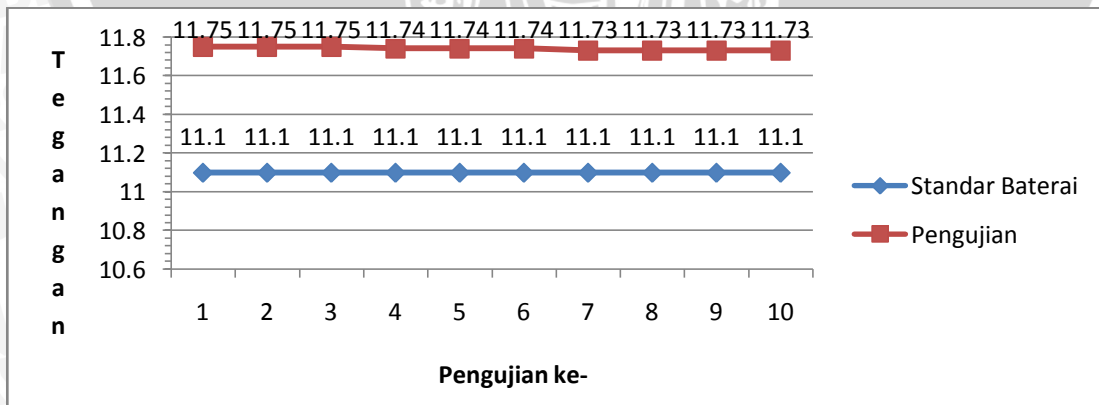
Pengujian										Rata-rata
1	2	3	4	5	6	7	8	9	10	
11.75	11.75	11.75	11.74	11.74	11.74	11.73	11.73	11.73	11.73	11.739

Pengujian catu tegangan baterai *Li-po* 11.1 V 2200 mA ditunjukkan dalam Gambar 5.1.



Gambar 5.1 Pengujian Catu Tegangan Baterai *Li-po* 11.1 V 2200 mA (Pengujian, 2013)

Grafik pengujian catu tegangan baterai *Li-po* 11.1 V 2200 mA ditunjukkan dalam Gambar 5.2.



Gambar 5.2 Grafik Catu Tegangan Baterai *Li-po* 11.1 V 2200 mA (Pengujian, 2013)

Dari hasil pengujian mengalami *error* sebesar:

$$\text{error}(\%) = \frac{|V_{\text{rata-rata}} - V_{\text{baterai}}|}{V_{\text{baterai}}} \times 100\% \dots\dots\dots (5.1)$$

dengan $V_{\text{rata-rata}}$ = tegangan rata-rata dari 10 kali pengujian

V_{baterai} = tegangan standar baterai 11.1 V

$$\text{error} = \frac{|11.739 - 11.1|}{11.1} \times 100\%$$

$$\text{error} = 5.76 \%$$

terjadi *error* sebesar 5.76 %, namun tegangan tersebut mampu memberikan catu lebih lama dalam pengujian.

5.2 Pengujian Tegangan Regulator dan ESC

Pengujian ini bertujuan untuk mengetahui hasil dari pengaturan tegangan regulator +5 V pada mikrokontroler dan tegangan keluaran pada rangkaian ESC.

5.2.1 Peralatan Pengujian

1. Baterai *Li-po* 11.1 V 2200A
2. Digital Multimeter SANWA CD772
3. Rangkaian regulator pada mikrokontroler
4. Rangkaian ESC FLAYFUN 30 A

5.2.2 Prosedur Pengujian

Pengujian dilakukan dengan menghubungkan baterai 11.1 V pada rangkaian regulator mikrokontroler. Kemudian tegangan keluaran diukur menggunakan multimeter digital dari regulator dan dilakukan pencatatan. Proses pengujian dilakukan sebanyak 10 kali. Untuk ESC dihubungkan dengan baterai 11.1 V pada kabel catu tegangan kemudian tegangan keluaran diukur menggunakan multimeter pada kabel tegangan keluaran ESC.

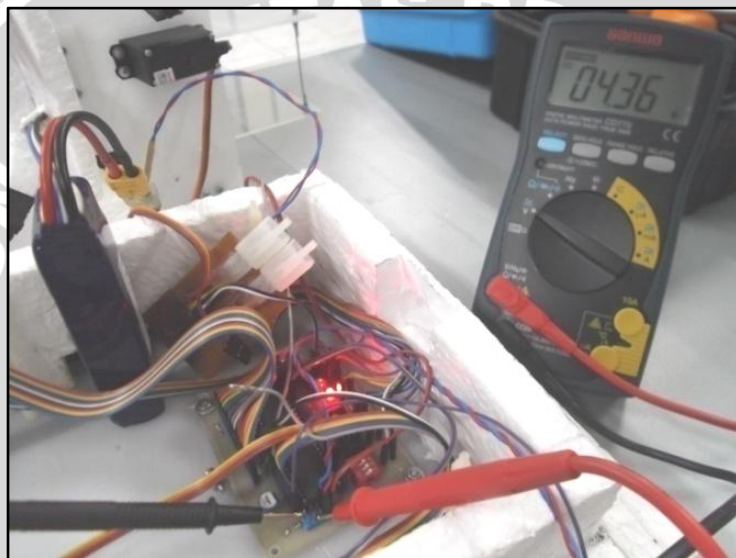
5.2.3 Hasil Pengujian

Setelah melakukan prosedur pengujian, maka didapatkan hasil untuk tegangan regulator dan ESC yang ditunjukkan dalam Tabel 5.2:

Tabel 5.2 Hasil Pengujian Tegangan Regulator dan ESC (Pengujian, 2013)

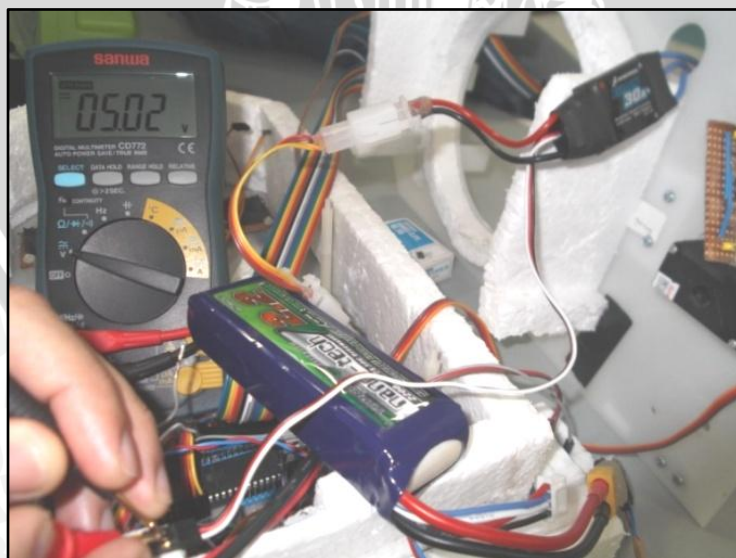
Pengujian										Rata-rata
1	2	3	4	5	6	7	8	9	10	
4.27	4.27	4.28	4.28	4.29	4.3	4.32	4.34	4.34	4.36	4.305
5.02	5.01	5.01	5.02	5.02	5.02	5	5	5.01	5.02	5.013

Proses pengujian untuk tegangan regulator ditunjukkan dalam Gambar 5.3.



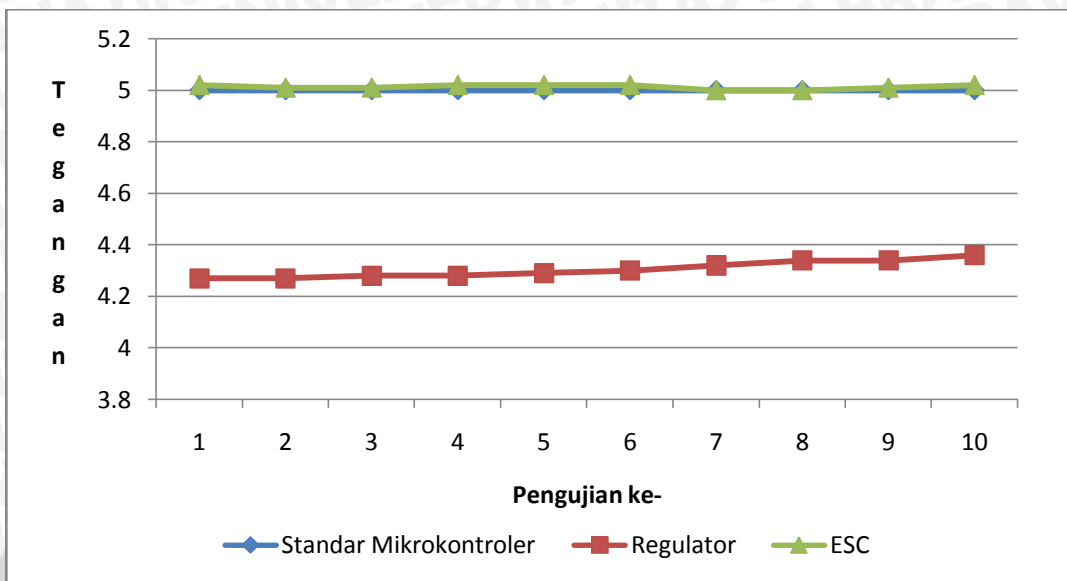
Gambar 5.3 Pengujian Tegangan Regulator +5 V (Pengujian, 2013)

Pengujian untuk tegangan ESC ditunjukkan dalam Gambar 5.4.



Gambar 5.4 Pengujian Tegangan ESC (Pengujian, 2013)

Grafik pengujian ditunjukkan dalam Gambar 5.5.



Gambar 5.5 Grafik Tegangan Regulator +5 V dan ESC (Pengujian, 2013)

Dari hasil pengujian tegangan regulator mengalami *error* sebesar:

$$error(\%) = \frac{|V_{rata-rata} - V_{MK}|}{V_{MK}} \times 100\% \dots\dots\dots (5.2)$$

dengan $V_{rata-rata}$ = tegangan rata-rata dari 10 kali pengujian

V_{MK} = tegangan standar mikrokontroler

$$error = \frac{|4.305 - 5|}{5} \times 100\%$$

$$error = 13.9 \%$$

Dan hasil pengujian tegangan ESC mengalami *error* sebesar:

$$error(\%) = \frac{|V_{rata-rata} - V_{MK}|}{V_{MK}} \times 100\% \dots\dots\dots (5.3)$$

dengan $V_{rata-rata}$ = tegangan rata-rata dari 10 kali pengujian

V_{MK} = tegangan standar mikrokontroler

$$error = \frac{|5.013 - 5|}{5} \times 100\%$$

$$error = 0.26 \%$$

Hasil dari pengujian regulator dan ESC menunjukkan bahwa *error* tegangan yang terjadi pada ESC lebih kecil daripada *error* tegangan pada regulator. Sehingga untuk memberi catu tegangan ke mikrokontroler menggunakan catu tegangan dari ESC dengan *error* 0.26 %.

5.3 Pengujian Tegangan Keluaran Sensor *Photo Reflective*

Pengujian ini bertujuan untuk mengetahui besar tegangan yang dihasilkan sensor saat membaca garis hitam. Besar kecilnya nilai tegangan akan memberikan masukan digital 0 atau 1 pada mikrokontroler.

5.3.1 Peralatan Pengujian

1. 8 sensor *photo reflective*
2. Catu tegangan dari ESC FLAYFUN 30 A
3. Rangkaian mikrokontroler
4. Baterai *Li-po* 11.1 V 2200mA
5. Digital Multimeter SANWA CD772

5.3.2 Prosedur Pengujian

Pengujian dilakukan dengan memberi catu tegangan pada ESC. Kemudian kabel keluaran 5 V ESC dihubungkan ke rangkaian paralel catu tegangan 5 V pada rangkaian mikrokontroler. Rangkaian sensor dihubungkan dengan catu tegangan pada mikrokontroler. Kemudian setiap sensor diarahkan pada garis hitam. Setiap kabel keluaran sensor diukur menggunakan multimeter untuk mengetahui tegangan yang dihasilkan oleh pembacaan sensor. Data yang diperoleh kemudian dicatat dan digunakan untuk menentukan nilai logika sensor pada mikrokontroler.

5.3.3 Hasil Pengujian

Setelah melakukan prosedur pengujian, didapatkan hasil yang ditunjukkan dalam tabel dan gambar.

Hasil pengujian sensor S0 mengenai garis hitam ditunjukkan dalam Tabel 5.3.

Tabel 5.3 Hasil Pengujian Sensor S0 Mengenai Garis Hitam (Pengujian, 2013)

S0	0	1	1	1	1	1	1	1
1	3.75	0.154	0.128	0.148	0.15	1.02	0.92	0.21
2	3.73	0.147	0.132	0.15	0.147	0.98	0.8	0.219
3	3.7	0.142	0.138	0.158	0.152	0.92	0.81	0.222
4	3.89	0.163	0.146	0.161	0.16	1.02	0.85	0.221
5	3.78	0.153	0.181	0.174	0.161	1.02	0.83	0.239
Rata-rata	3.77	0.152	0.145	0.158	0.154	0.992	0.842	0.222

Hasil pengujian sensor S1 mengenai garis hitam ditunjukkan dalam Tabel 5.4.

Tabel 5.4 Hasil Pengujian Sensor S1 Mengenai Garis Hitam (Pengujian, 2013)

S1	1	0	1	1	1	1	1	1
1	0.189	3.96	0.12	0.138	0.131	0.145	0.2	0.22
2	0.188	3.97	0.122	0.137	0.133	0.185	0.228	0.222
3	0.191	3.8	0.123	0.138	0.136	0.229	0.225	0.222
4	0.19	3.86	0.123	0.137	0.135	0.217	0.22	0.225
5	0.191	3.96	0.123	0.138	0.136	0.219	0.211	0.222
Rata-rata	0.19	3.91	0.122	0.138	0.134	0.199	0.217	0.222

Hasil pengujian sensor S2 mengenai garis hitam ditunjukkan dalam Tabel 5.5.

Tabel 5.5 Hasil Pengujian Sensor S2 Mengenai Garis Hitam (Pengujian, 2013)

S2	1	1	0	1	1	1	1	1
1	0.224	0.188	4.48	0.176	0.165	0.233	0.685	0.233
2	0.21	0.204	4.49	0.159	0.149	0.224	0.746	0.245
3	0.217	0.184	4.52	0.154	0.139	0.222	0.7	0.228
4	0.209	0.181	4.45	0.152	0.142	0.219	0.655	0.229
5	0.209	0.181	4.5	0.153	0.141	0.225	0.629	0.223
Rata-rata	0.214	0.188	4.488	0.159	0.147	0.225	0.683	0.232

Hasil pengujian sensor S3 mengenai garis hitam ditunjukkan dalam Tabel 5.6.

Tabel 5.6 Hasil Pengujian Sensor S3 Mengenai Garis Hitam (Pengujian, 2013)

S3	1	1	1	0	1	1	1	1
1	0.192	0.15	0.17	4.45	0.139	0.18	0.186	0.222
2	0.203	0.146	0.166	4.46	0.141	0.187	0.187	0.226
3	0.195	0.147	0.164	4.48	0.143	0.187	0.188	0.222
4	0.239	0.141	0.154	4.46	0.136	0.181	0.177	0.225
5	0.194	0.135	0.15	4.47	0.132	0.183	0.18	0.229
Rata-rata	0.205	0.144	0.161	4.464	0.138	0.184	0.184	0.225

Hasil pengujian sensor S4 mengenai garis hitam ditunjukkan dalam Tabel 5.7.

Tabel 5.7 Hasil Pengujian Sensor S4 Mengenai Garis Hitam (Pengujian, 2013)

S4	1	1	1	1	0	1	1	1
1	0.2	0.13	0.129	0.162	4.49	0.146	0.133	0.221
2	0.199	0.135	0.133	0.172	4.38	0.148	0.136	0.222
3	0.235	0.123	0.122	0.175	4.48	0.137	0.123	0.23
4	0.201	0.129	0.126	0.191	4.38	0.142	0.128	0.229
5	0.198	0.13	0.127	0.182	4.38	0.148	0.14	0.228
Rata-rata	0.207	0.129	0.127	0.176	4.422	0.144	0.132	0.226

Hasil pengujian sensor S5 mengenai garis hitam ditunjukkan dalam Tabel 5.8.

Tabel 5.8 Hasil Pengujian Sensor S5 Mengenai Garis Hitam (Pengujian, 2013)

S5	1	1	1	1	1	0	1	1
1	0.198	0.134	0.122	0.159	0.173	4.63	0.93	0.225
2	0.195	0.137	0.126	0.164	0.179	4.53	0.89	0.224
3	0.195	0.136	0.126	0.165	0.185	4.66	1.04	0.239
4	0.191	0.139	0.126	0.167	0.19	4.53	1	0.222
5	0.201	0.137	0.125	0.163	0.179	4.52	0.91	0.225
Rata-rata	0.196	0.137	0.125	0.164	0.181	4.574	0.954	0.227

Hasil pengujian sensor S6 mengenai garis hitam ditunjukkan dalam Tabel 5.9.

Tabel 5.9 Hasil Pengujian Sensor S6 Mengenai Garis Hitam (Pengujian, 2013)

S6	1	1	1	1	1	1	0	1
1	0.205	0.139	0.131	0.14	0.151	0.632	4.7	0.234
2	0.206	0.144	0.142	0.145	0.142	0.69	4.59	0.232
3	0.21	0.147	0.143	0.147	0.143	0.526	4.58	0.23
4	0.211	0.143	0.138	0.15	0.152	0.41	4.59	0.229
5	0.201	0.144	0.141	0.153	0.144	0.358	4.57	0.237
Rata-rata	0.207	0.143	0.139	0.147	0.146	0.523	4.606	0.232

Hasil pengujian sensor S7 mengenai garis hitam ditunjukkan dalam Tabel 5.10.

Tabel 5.10 Hasil Pengujian Sensor S7 Mengenai Garis Hitam (Pengujian, 2013)

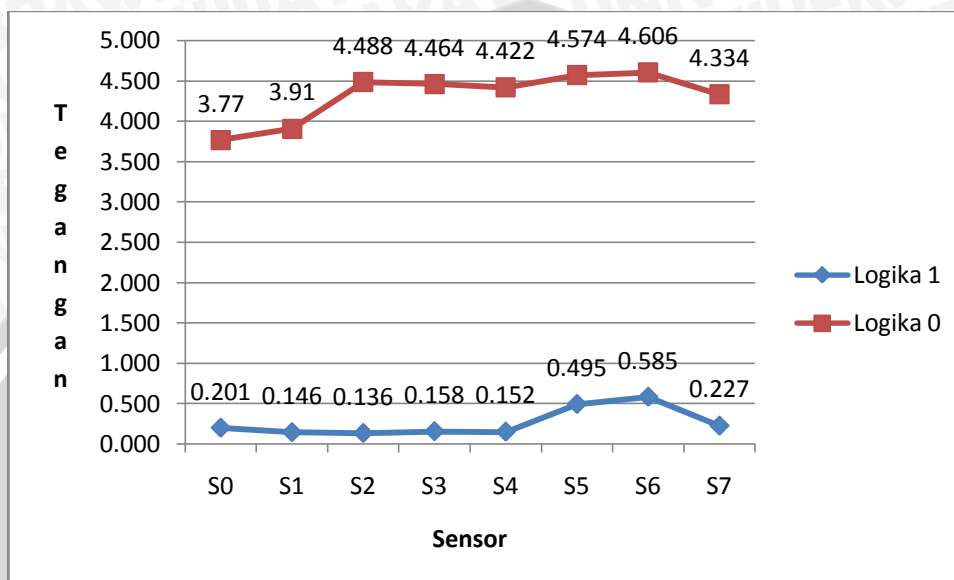
S7	1	1	1	1	1	1	1	0
1	0.201	0.13	0.123	0.153	0.156	1.061	1.051	4.39
2	0.191	0.128	0.131	0.163	0.158	1.101	1.08	4.24
3	0.191	0.129	0.132	0.166	0.161	1.129	1.086	4.4
4	0.186	0.127	0.13	0.173	0.166	1.314	1.096	4.29
5	0.192	0.126	0.131	0.176	0.175	1.394	1.112	4.35
Rata-rata	0.192	0.128	0.129	0.166	0.163	1.2	1.085	4.334

Dari pengujian diambil rata-rata nilai tegangan pada saat logika 0 (sensor mendeteksi warna hitam) dan pada saat logika 1 (sensor mendeteksi warna putih) pada tiap sensor yang ditunjukkan dalam Tabel 5.11.

Tabel 5.11 Rata-rata Nilai Tegangan pada Logika Tiap Sensor (Pengujian, 2013)

Sensor	Logika	
	1 (Putih)	0 (Hitam)
S0	0.201	3.77
S1	0.146	3.91
S2	0.136	4.488
S3	0.158	4.464
S4	0.152	4.422
S5	0.495	4.574
S6	0.585	4.606
S7	0.227	4.334

Penentuan tegangan untuk logika 0 dan 1 diambil dari jarak minimum dan maksimum tegangan yang terukur. Sehingga ditentukan rentang tegangan untuk logika 1 adalah 0 – 1 V dan logika 0 adalah 3.5 – 5 V. Gambar 5.6 menunjukkan grafik rata-rata tegangan logika tiap sensor.



Gambar 5.6 Grafik Rata-rata Nilai Tegangan pada Logika Tiap Sensor (Pengujian, 2013)

5.4 Pengujian RPM dan *Thrust* Motor DC *Brushless Hover*

Pengujian ini bertujuan untuk mengetahui RPM konstan dan gaya *thrust* yang terjadi pada saat robot mengalami *hover*.

5.4.1 Peralatan Pengujian

1. Motor DC *Brushless* C2822-1400kV
2. *Propeller* EMP 6x5E
3. ESC FLAYFUN 30 A
4. Rangkaian mikrokontroler
5. Timbangan YAMATO 3 kg
6. *Digital Tachometer* DT-2234B
7. Baterai *Li-po* 11.1 V 2200A

5.4.2 Prosedur Pengujian

Pengujian dilakukan dengan merangkai motor DC *brushless*, *propeller*, ESC, dan rangkaian mikrokontroler sesuai dengan PIN program untuk motor DC *brushless*. Seluruh rangkaian diposisikan pada timbangan. Timbangan berfungsi sebagai pengujian tekanan dorongan dan ditunjukkan dalam massa. Kemudian catu tegangan diaktifkan. Saat putaran motor konstan, RPM diukur menggunakan *digital tachometer* dan massa yang terukur timbangan dicatat. Proses tersebut diulang sebanyak 10 kali untuk mendapatkan data yang lebih detail.

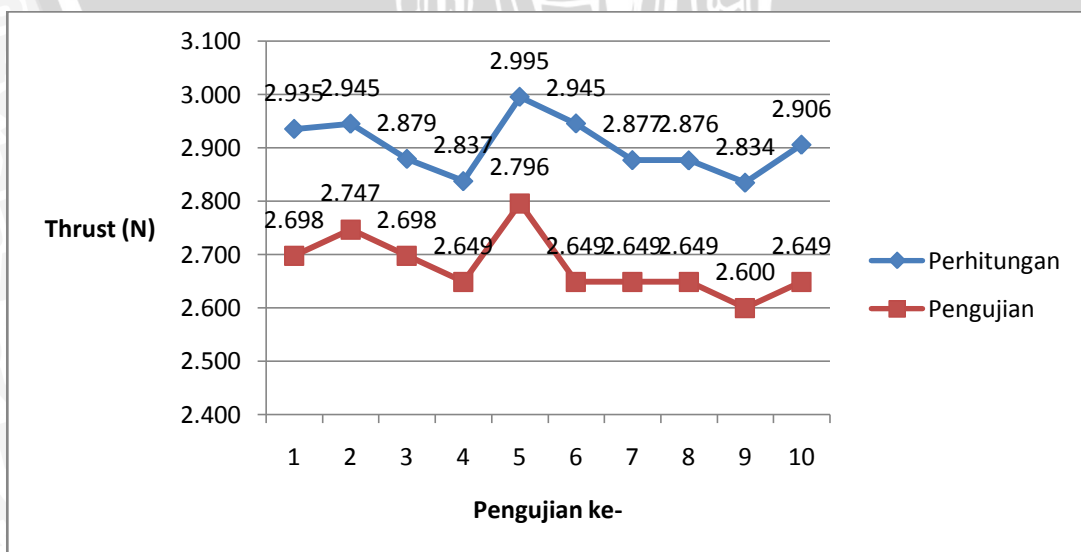
5.4.3 Hasil Pengujian

Setelah melakukan prosedur pengujian, didapatkan hasil yang ditunjukkan dalam Tabel 5.12.

Tabel 5.12 Hasil Pengujian RPM dan *Thrust* Motor DC *Brushless Hover* (Pengujian, 2013)

No.	RPM	massa (kg)	ω (rad/s)	Thrust (N) Perhitungan	Thrust (N) Pengujian
1	12068	0.275	1264.284	2.935	2.698
2	12088	0.28	1266.379	2.945	2.747
3	11952	0.275	1252.131	2.879	2.698
4	11865	0.27	1243.017	2.837	2.649
5	12191	0.285	1277.170	2.995	2.796
6	12089	0.27	1266.484	2.945	2.649
7	11947	0.27	1251.608	2.877	2.649
8	11946	0.27	1251.503	2.876	2.649
9	11859	0.265	1242.388	2.834	2.600
10	12007	0.27	1257.893	2.906	2.649

Grafik *thrust* motor DC *brushless hover* ditunjukkan dalam Gambar 5.7.



Gambar 5.7 Grafik *Thrust* Motor DC *Brushless Hover* (Pengujian, 2013)

Dari hasil pengujian, *thrust* mengalami *error* sebesar:

$$error(\%) = \frac{|Thrust_U - Thrust_H|}{Thrust_H} \times 100\% \dots\dots\dots (5.4)$$

dengan $Thrust_U$ = gaya rata-rata dari 10 kali pengujian

$Thrust_H$ = gaya dari perhitungan (RPM rata-rata pengujian 12001.2)

$$error = \frac{|2.678 - 2.903|}{2.903} \times 100\%$$

$$error = 7.751 \%$$

5.5 Pengujian Sinyal PWM Motor Servo

Pengujian ini bertujuan untuk mengetahui bentuk sinyal saat pada posisi sudut-sudut yang telah ditentukan untuk mengarahkan hembusan angin motor DC *brushless* pendorong.

5.5.1 Peralatan Pengujian

1. Baterai *Li-po* 11.1 V 2200A
2. Rangkaian mikrokontroler
3. ESC FLAYFUN 30 A
4. PC-SCOPE PCSU1000 1Gs/s – 2x60 MHz
5. Komputer PC dan program *software* PCLab2000SE v4.04

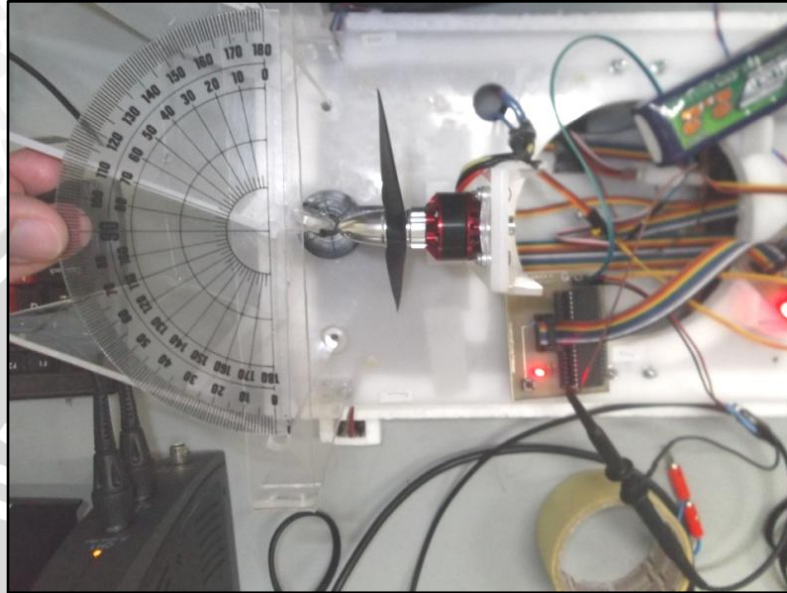
5.5.2 Prosedur Pengujian

Pengujian dilakukan dengan menghubungkan baterai dengan ESC untuk memberi tegangan rangkaian mikrokontroler. Kemudian menghubungkan PIND.7 sebagai keluaran pada motor servo dengan konektor PC-SCOPE. Kemudian dibuka *software* PCLab2000SE pada PC dan dipilih *mode oscilloscope*. Setelah semua terhubung, baterai di-*on*-kan, dan *oscilloscope* dijalankan. Setelah gambar sinyal muncul data disimpan. Dan dilakukan pengujian sinyal pada tiap-tiap sudut yang dipakai yaitu 55^0 , 70^0 , 80^0 , 90^0 , 100^0 , 110^0 , dan 125^0 .

5.5.3 Hasil Pengujian

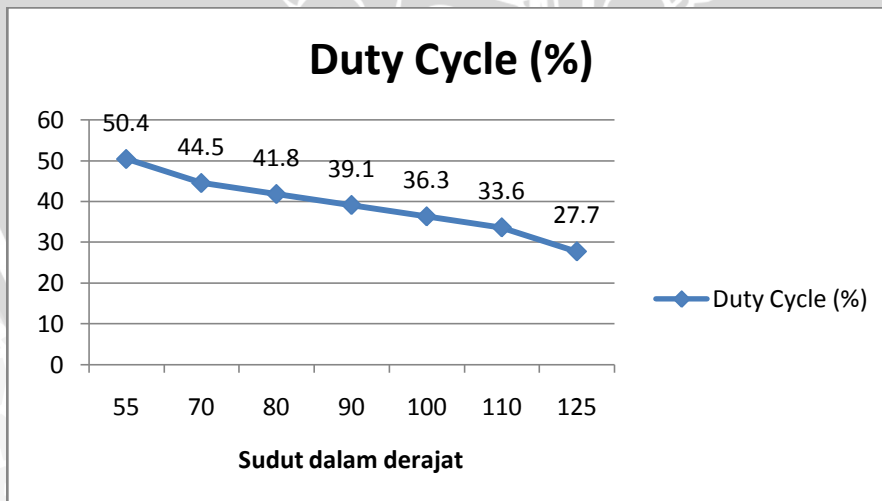
Setelah melakukan prosedur pengujian, didapatkan hasil untuk sinyal dan tampilan parameter. Parameter yang ditunjukkan adalah amplitudo, *duty cycle*, lebar

sinyal saat *on* dan *off*, periode lebar sinyal, dan frekuensi. Gambar untuk sinyal dan parameter dapat dilihat pada halaman lampiran. Pengujian sudut servo ditunjukkan dalam Gambar 5.8.



Gambar 5.8 Pengujian Sudut Motor Servo pada Sudut 110⁰ (Pengujian, 2013)

Grafik *duty cycle* ditunjukkan dalam Gambar 5.9.



Gambar 5.9 Grafik *Duty Cycle* Motor Servo (pengujian, 2013)

Dari grafik terlihat nilai-nilai *duty cycle* dalam sudut-sudut yang digunakan untuk menggerakkan motor DC servo pada sirip/ kendali arah angin motor DC *brushless* pendorong.

5.6 Pengujian Massa Robot *Hovercraft Line Follower*

Pengujian ini bertujuan untuk mengetahui massa robot secara keseluruhan saat *OFF* dan *ON*. Fungsi dari massa adalah sebagai nilai yang digunakan untuk mengetahui gaya berat dari robot.

5.6.1 Peralatan Pengujian

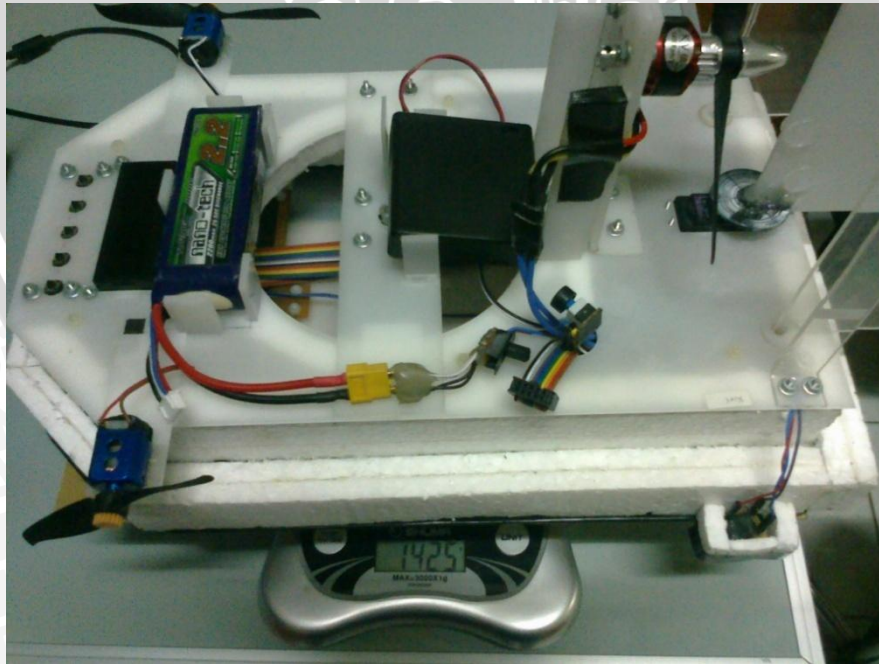
1. Robot *Hovercraft Line Follower*
2. Timbangan SHUMA Digital Kitchen Scale SHDK3000

5.6.2 Prosedur Pengujian

Pengujian dilakukan dengan meletakkan robot *hovercraft line follower* di atas timbangan digital. Kemudian dicatat massa yang tertunjuk oleh jarum pengukur. Pengujian dilakukan saat kondisi robot *OFF* dan kondisi robot *ON* kemudian diambil data massa robot.

5.6.3 Hasil Pengujian

Setelah melakukan prosedur pengujian, didapatkan hasil massa robot *hovercraft line follower* sebesar 1.425 kg saat *OFF* yang ditunjukkan dalam Gambar 5.10.



Gambar 5.10 Pengujian Massa Robot *Hovercraft Line Follower* saat *OFF* (Pengujian, 2013)

Hasil pengujian massa robot *hovercraft line follower* sesaat bernilai sebesar 1.390 kg kemudian kembali ke nilai 1.425 kg saat *ON* dengan kondisi melayang ditunjukkan dalam Gambar 5.11.



Gambar 5.11 Pengujian Massa Robot *Hovercraft Line Follower* saat *ON* (Pengujian, 2013)

Gaya berat yang terjadi saat *OFF* adalah:

$$F = ma \dots\dots\dots (5.5)$$

dengan F = gaya berat (N)
 m = massa benda (kg)
 a = gravitasi bumi (9.81 m/s^2)

maka:
 $F = 1.425 \text{ kg} \times 9.81 \text{ m/s}^2$
 $= 13.979 \text{ N}$

dan gaya berat yang terjadi saat *ON* adalah:

$$F = ma \dots\dots\dots (5.6)$$

dengan F = gaya berat (N)
 m = massa benda (kg)
 a = gravitasi bumi (9.81 m/s^2)

maka:
 $F = 1.39 \text{ kg} \times 9.81 \text{ m/s}^2$
 $= 13.636 \text{ N}$

Dari pengujian massa robot *hovercraft line follower* saat *OFF* dan *ON* didapatkan selisih sesaat massa sebesar 0.035 kg dan selisih sesaat gaya berat 0.343 N kemudian timbangan menunjukkan nilai massa yang sama yaitu 1.425 kg dengan selisih gaya = 0.



5.7 Pengujian Nilai Kcr

Pengujian ini bertujuan untuk mendapatkan nilai kritis Kcr dan respon sistem osilasi yang berkelanjutan.

5.7.1 Peralatan Pengujian

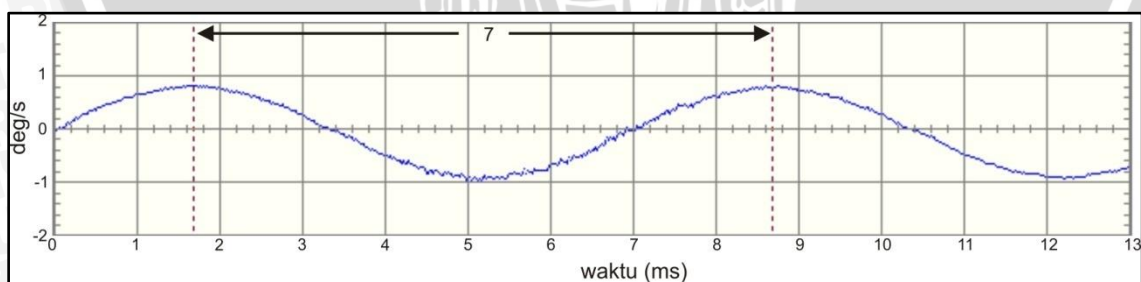
1. Robot *hovercraft line follower*
2. PC-SCOPE PCSU1000 1Gs/s – 2x60 MHz
3. Komputer PC dan program *software* PCLab2000SE v4.04

5.7.2 Prosedur Pengujian

Pengujian dilakukan dengan menghubungkan konektor PC-SCOPE dengan motor DC *brushless* pendorong. Kemudian dibuka *software* PCLab2000SE pada PC diaktifkan dan dipilih mode *oscilloscope*. Setelah semua terhubung, *oscilloscope* dijalankan, kemudian robot diaktifkan dengan diberikan nilai Kp. Sinyal yang muncul disimpan dan dilakukan peningkatan nilai Kp sampai ditemukan nilai kritis Kcr dengan sinyal osilasi berkesinambungan yang dibutuhkan.

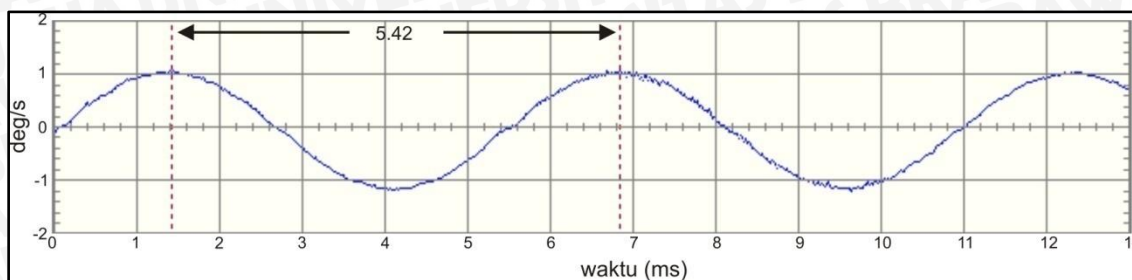
5.7.3 Hasil Pengujian

Setelah melakukan prosedur pengujian, didapatkan hasil sinyal pada kenaikan nilai Kp hingga didapatkan sinyal nilai kritis Kcr. Sinyal yang diperoleh untuk nilai Kp = 14 ditunjukkan dalam Gambar 5.12.



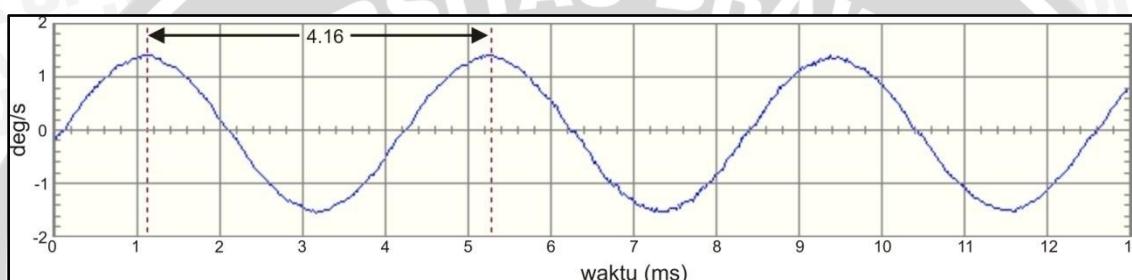
Gambar 5.12 Sinyal Osilasi Berkesinambungan untuk Nilai Kp = 14 (Pengujian, 2013)

Sinyal yang diperoleh untuk nilai $K_p = 16$ ditunjukkan dalam Gambar 5.13.



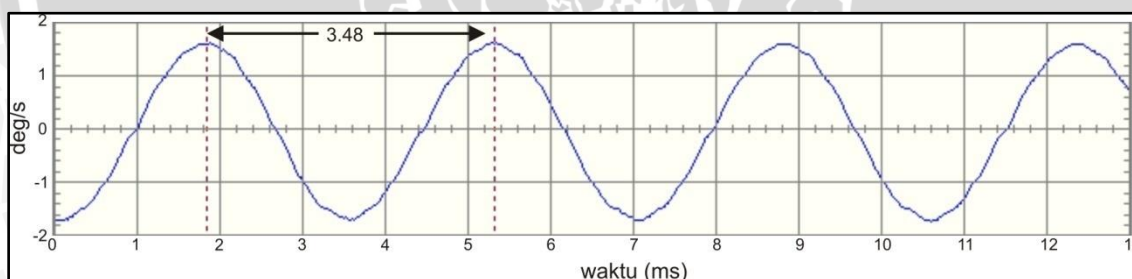
Gambar 5.13 Sinyal Osilasi Berkesinambungan untuk Nilai $K_p = 16$ (Pengujian, 2013)

Sinyal yang diperoleh untuk nilai $K_p = 18$ ditunjukkan dalam Gambar 5.14.



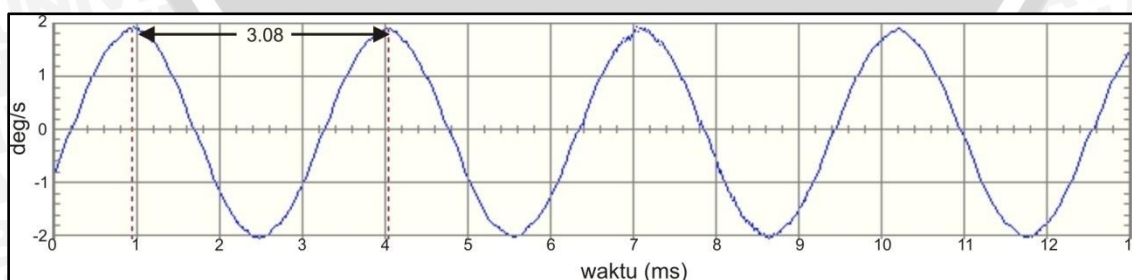
Gambar 5.14 Sinyal Osilasi Berkesinambungan untuk Nilai $K_p = 18$ (Pengujian, 2013)

Sinyal yang diperoleh untuk nilai $K_p = 20$ ditunjukkan dalam Gambar 5.15.



Gambar 5.15 Sinyal Osilasi Berkesinambungan untuk Nilai $K_p = 20$ (Pengujian, 2013)

Sinyal yang diperoleh untuk nilai $K_p = 22$ ditunjukkan dalam Gambar 5.16.



Gambar 5.16 Sinyal Osilasi Berkesinambungan untuk Nilai $K_p = 22$ (Pengujian, 2013)

Hasil dari nilai kritis Kcr yang sesuai dengan kondisi respon robot *hovercraft line follower* untuk dapat dikontrol pada nilai kritis Kcr = 20.

5.8 Pengujian RPM Motor DC *Brushless* Pendorong

Pengujian ini bertujuan untuk mengetahui kesesuaian program dengan kondisi sebenarnya saat robot dijalankan. Perubahan kecepatan akan terjadi saat mengikuti garis hitam yang memiliki pola belokan.

5.8.1 Peralatan Pengujian

1. Robot *hovercraft line follower*
2. Lintasan bergaris
3. *Digital Tachometer* DT-2234B

5.8.2 Prosedur Pengujian

Pengujian ini dilakukan dengan mengaktifkan robot *hovercraft line follower* dan memposisikan pada garis hitam. Robot diposisikan pada saat kondisi *steady* dan diukur RPM yang terjadi dengan *digital tachometer*. Kemudian dilakukan gangguan berupa pergeseran robot menjauhi garis *steady* dan diukur kembali RPM yang terjadi.

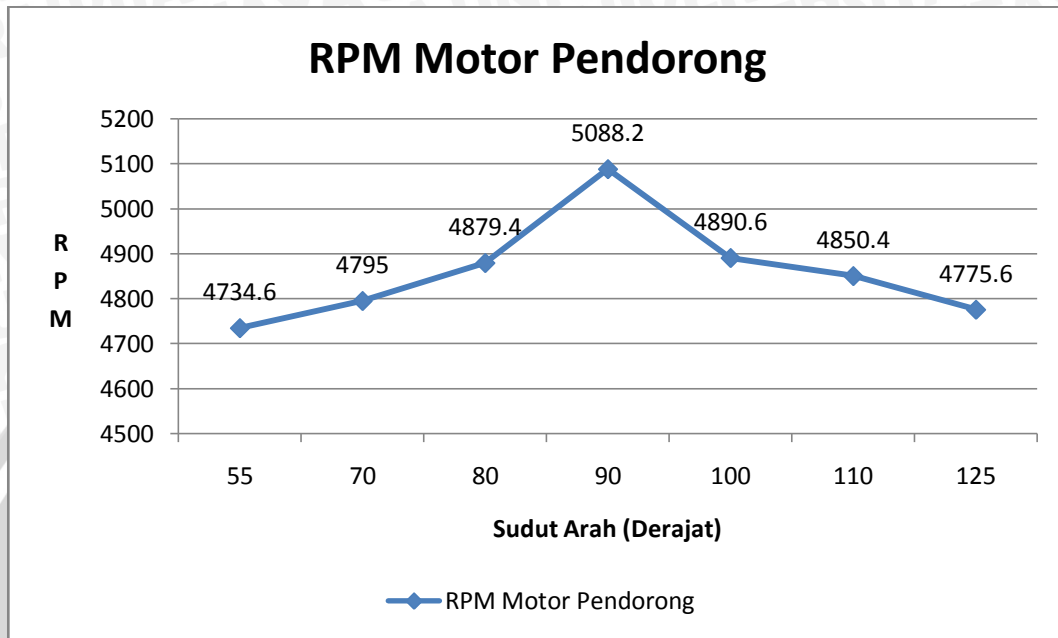
5.8.3 Hasil Pengujian

Setelah melakukan prosedur pengujian, didapatkan hasil yang ditunjukkan dalam Tabel 5.13.

Tabel 5.13 Pengujian RPM Motor DC *Brushless* Pendorong (Pengujian, 2013)

Arah	Sudut (derajat)	Pengujian (RPM)					Rata-rata (RPM)
		1	2	3	4	5	
kasudut3	55	4755	4769	4713	4738	4698	4734.6
kasudut2	70	4824	4801	4798	4787	4765	4795
kasudut1	80	4895	4853	4877	4905	4867	4879.4
Lurus	90	5180	5093	4989	5134	5045	5088.2
kisudut1	100	4921	4905	4893	4855	4879	4890.6
kisudut2	110	4883	4853	4871	4848	4797	4850.4
kisudut3	125	4780	4797	4751	4762	4788	4775.6

Gambar 5.17 menunjukkan perubahan RPM saat kondisi sensor berada pada saat *steady* dan saat ada gangguan yaitu garis belok.



Gambar 5.17 Grafik RPM Motor DC *Brushless* Pendorong (Pengujian, 2013)

Hasil yang diperoleh dari pengujian RPM pada saat sensor robot diarahkan pada garis lurus dan sirip membentuk sudut 90^0 . Kemudian digeser secara perlahan menjauhi garis ke kanan dan kekiri. Kecepatan motor DC *brushless* pendorong mengalami penurunan sesuai harapan.

5.9 Pengujian Sistem Keseluruhan

Pengujian ini bertujuan untuk melihat respon posisi robot secara keseluruhan saat mengikuti garis hitam dengan jalur yang telah ditentukan.

5.9.1 Peralatan Pengujian

1. Robot *hovercraft line follower*
2. Lintasan bergaris
3. Mistar *STAINLESS*

5.9.2 Prosedur Pengujian

Pengujian ini dilakukan dengan mengaktifkan robot yang telah diprogram pada jalur lurus garis hitam. kemudian diamati simpangan terjauh robot yang terjadi saat mengikuti garis lurus.

5.9.3 Hasil Pengujian

Setelah melakukan prosedur pengujian didapatkan hasil yang ditunjukkan dalam Tabel 5.14.

Tabel 5.14 Hasil Pengujian Sistem Keseluruhan (Pengujian, 2013)

Jarak (cm)	Error (cm)	Jarak (cm)	Error (cm)	Jarak (cm)	Error (cm)	Jarak (cm)	Error (cm)
1	-0.5	26	0	51	0	76	0
2	-1	27	0	52	0	77	0
3	-1.5	28	0	53	0	78	0
4	-1	29	0	54	0	79	0
5	0	30	0	55	0	80	0
6	0	31	0	56	0	81	-1
7	0	32	0	57	0	82	-1
8	-1	33	0	58	0	83	-1
9	-1	34	0	59	0	84	-1
10	0	35	0	60	0	85	-1
11	0	36	0	61	0	86	-1
12	0	37	0	62	0	87	-1
13	0	38	0	63	0	88	-1
14	0	39	0	64	0	89	-1
15	1	40	0	65	0	90	-1
16	1	41	0	66	0	91	-1
17	1	42	0	67	0	92	-1
18	1	43	0	68	0	93	0
19	0	44	0	69	0	94	1
20	0	45	0	70	0	95	1
21	0	46	0	71	0	96	1
22	0	47	0	72	0	97	1
23	0	48	0	73	0	98	2
24	0	49	0	74	0	99	3
25	0	50	0	75	0	100	3

Dari hasil didapatkan beberapa pergeseran yang merupakan *error*. Untuk nilai pergeseran = -0.5 cm, sensor yang mengenai garis hitam = S3, dengan nilai pembebanan PV = 1 maka:

$error = set_point - PV$ (set_point = 0 adalah posisi lurus robot saat mendeteksi garis)

$$error = 0 - 1$$

$$error = -1$$

$$nil_pid = ((Kp*(-error))+((Ki*((-error)+error1)*Ts))+((Kd/Ts)*((-error)-error1)))$$

dengan kondisi

jika $0 \leq \text{nil_pid} < 50$ maka memanggil nilai = 0

jika $50 \leq \text{nil_pid} < 100$ maka memanggil nilai = -1

jika $150 \leq \text{nil_pid} < 235$ maka memanggil nilai = -2

jika $240 \leq \text{nil_pid} < 400$ maka memanggil nilai = -3

$\text{error1} = (-\text{error})$

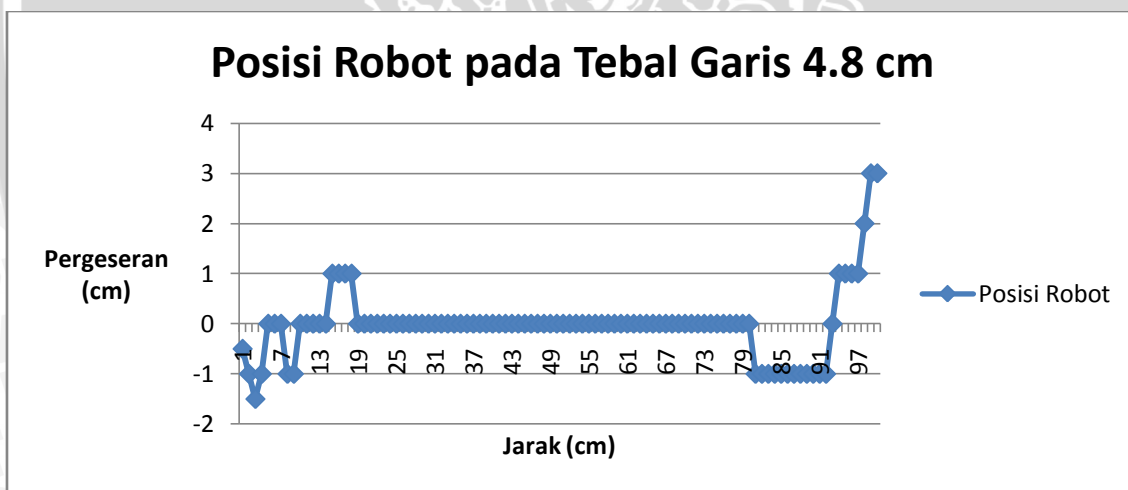
$\text{nil_pid} = ((12 * (-(-1))) + (6.897 * ((-(-1)) + 0) * 10) + ((5.22/10) * ((-(-1)) - 0)))$

$\text{nil_pid} = 12 + 68.97 + 0.522$

$\text{nil_pid} = 81.492$

nilai kontrol PID yang diperoleh adalah 81.492, maka pemanggilan yang terjadi untuk mengubah nilai PWM motor DC *brushless* pendorong sesuai kondisi robot adalah -1.

Grafik pengujian sistem keseluruhan ditunjukkan dalam Gambar 5.18.



Gambar 5.18 Grafik Pengujian Sistem Keseluruhan (Pengujian, 2013)

Dari grafik didapatkan *error* pergeseran terjauh sebesar:

$$\text{error}_{\max}(\%) = \frac{|x_{\max}|}{x_s} \times 100\% \dots\dots\dots (5.7)$$

dengan x_{\max} = pergeseran terjauh

x_s = posisi sensor *setpoint* dari sensor pinggir

$$\text{error}_{\max} = \frac{|3|}{14} \times 100\%$$

$$\text{error}_{\max} = 21.43 \%$$



Dan didapatkan *error* pergeseran terdekat sebesar:

$$error_{min}(\%) = \frac{|x_{min}|}{x_s} \times 100\% \dots\dots\dots (5.8)$$

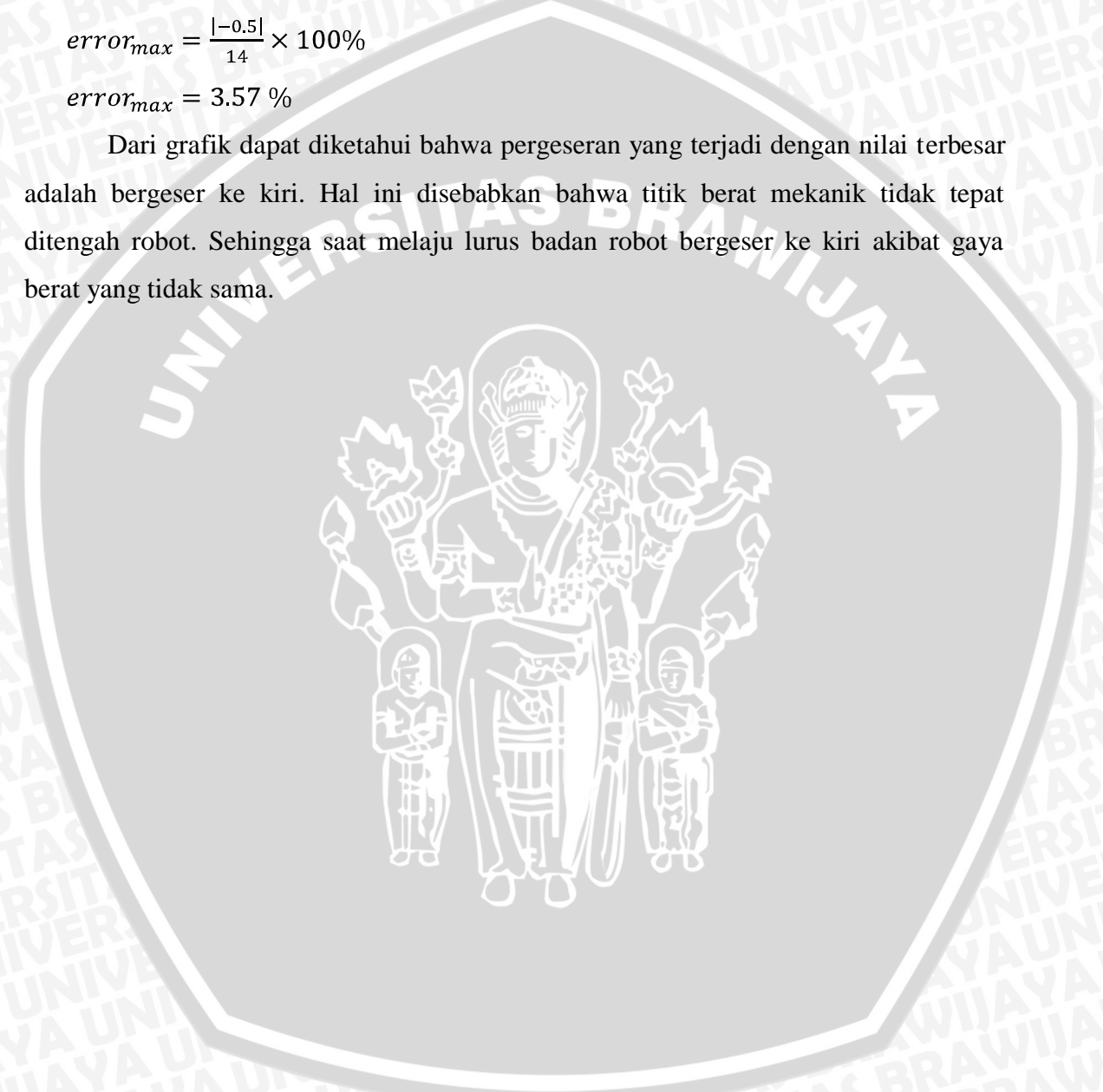
dengan x_{min} = pergeseran terdekat

x_s = posisi sensor *setpoint* dari sensor pinggir

$$error_{max} = \frac{|-0.5|}{14} \times 100\%$$

$$error_{max} = 3.57 \%$$

Dari grafik dapat diketahui bahwa pergeseran yang terjadi dengan nilai terbesar adalah bergeser ke kiri. Hal ini disebabkan bahwa titik berat mekanik tidak tepat ditengah robot. Sehingga saat melaju lurus badan robot bergeser ke kiri akibat gaya berat yang tidak sama.



BAB VI PENUTUP

6.1 Kesimpulan

Dari perancangan, pengujian, dan pengamatan yang telah dilakukan pada robot *hovercraft line follower*, maka dapat diambil kesimpulan sebagai berikut:

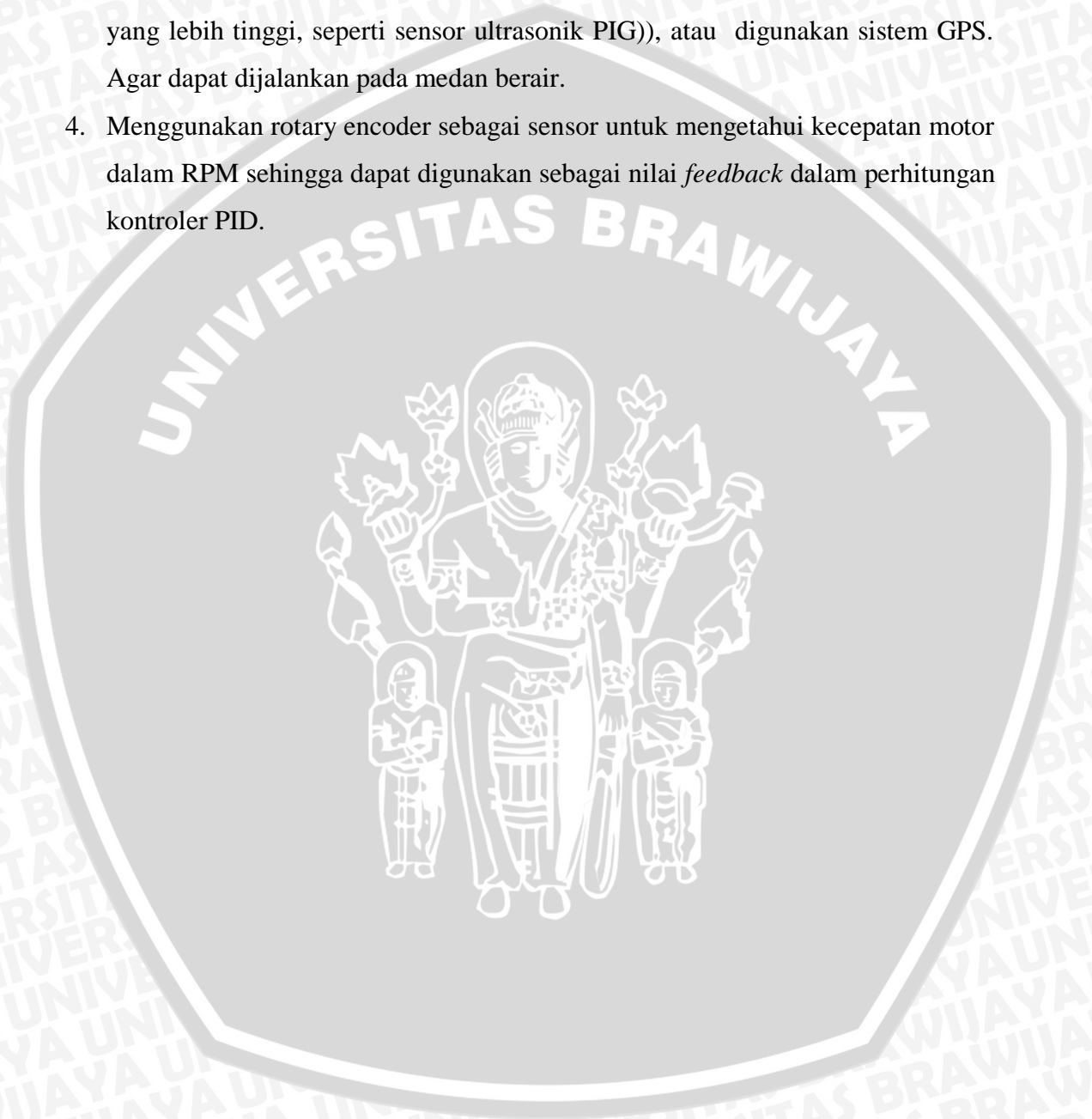
1. Dimensi robot *hovercraft line follower* memiliki panjang 45 cm, lebar 28 cm, dan tinggi 24 cm pada saat keadaan *off*. Robot *hovercraft line follower* menggunakan 8 sensor *photo reflective* sebagai *feedback* sistem. Satu motor servo dan dua motor DC magnet permanen sebagai sistem kendali arah. Dua motor DC *brushless* 1400 kV sebagai aktuator. Gaya angkat untuk *hover* dihasilkan melalui perpaduan kecepatan putaran motor DC *brushless* dan *propeller* EMP 6x5E dengan gaya angkat rata-rata mencapai 2.678 N pada kecepatan rata-rata 12001.2 RPM. Kecepatan rata-rata motor DC *brushless* pendorong robot *hovercraft line follower* saat berada pada garis lurus mencapai kecepatan 5088.2 RPM dengan gaya dorong sebesar 0.522 N.
2. Dengan menggunakan algoritma PID, sistem robot *hovercraft line follower* mampu menaikkan/ menurunkan kecepatan motor DC *brushless* pendorong saat mengikuti garis hitam dengan panjang 1 m dan lebar 4.8 cm. parameter PID ditentukan dengan menggunakan metode kedua Ziegler-Nichols dan didapatkan nilai $K_p = 12$, $K_i = 6.897$, dan $K_d = 5.22$. Dengan menggunakan parameter tersebut sistem robot *hovercraft line follower* memiliki respon sistem yang mampu mempertahankan posisi setimbang saat mengikuti garis.

6.2 Saran

Dalam perancangan dan pembuatan alat ini masih terdapat beberapa kelemahan. Untuk memperbaiki kinerja robot *hovercraft line follower* dan pengembangan lebih lanjut disarankan:

1. Penyempurnaan mekanik dengan lebih teliti dalam pembagian porsi beban kanan dan kiri robot. Beban yang tidak setimbang berpengaruh besar pada laju stabil robot saat posisi melayang/ *hover*.

2. Penambahan *logger* agar mampu merekam data pada saat *hover* dan melaju mengikuti garis, sehingga dapat menganalisis sistem robor *hovercraft line follower*.
3. Mengganti sensor *photo reflective* dengan sensor yang memiliki kemampuan yang lebih tinggi, seperti sensor ultrasonik (PIG)), atau digunakan sistem GPS. Agar dapat dijalankan pada medan berair.
4. Menggunakan rotary encoder sebagai sensor untuk mengetahui kecepatan motor dalam RPM sehingga dapat digunakan sebagai nilai *feedback* dalam perhitungan kontroler PID.



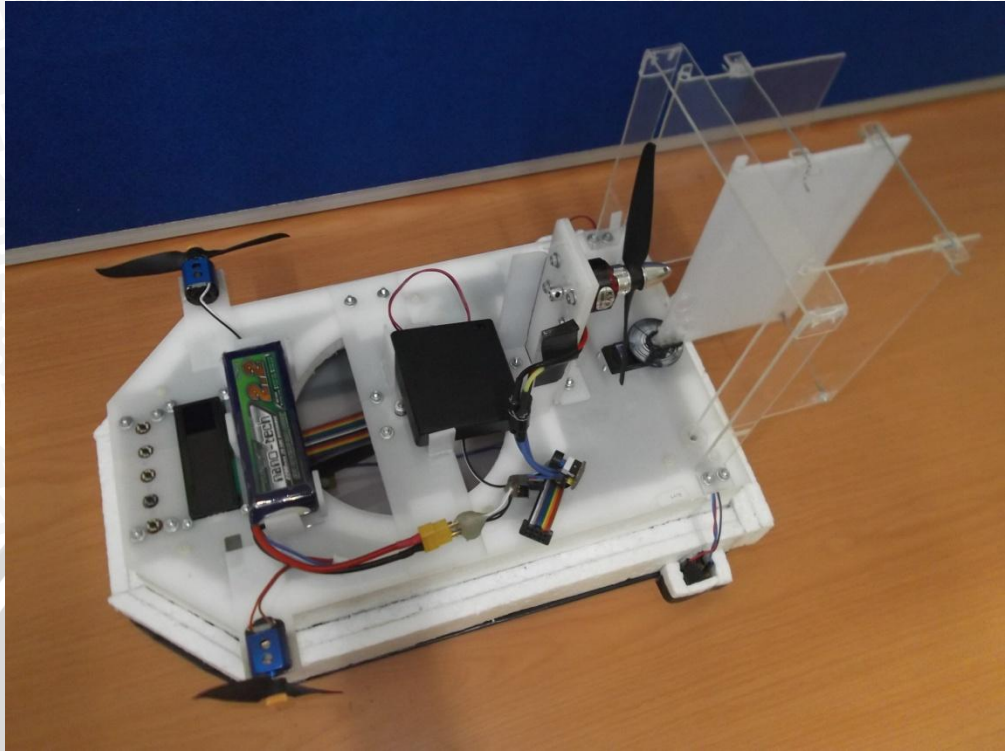
DAFTAR PUSTAKA

- Astrom, K. J., & Hagglund, Tore. 1995. *PID Controllers: Theory, Design and Tuning*. Research Triangle Park: Instrument Society of America.
- Atmel Corporation. 2010. *ATMEGA 8535 series*.
- Brown, Ward. 2002. *Brushless DC Motor Control Made Easy*. Microchip Technology, Inc.
- Buchi, Roland. 2012. *Brushless Motors and Controllers*. Norderstedt. Books on Demand.
- Dwi Septian Suyadhi, Taufiq. 2010. Buku Pintar Robotika. Yogyakarta: ANDI.
- Dwi Yuliawan, Hendra. 2010. Perencanaan Sistem Thruster dan Lifter Tipe Terpisah untuk Hovercraft Militer dengan Payload 15 ton. Proceedings Tugas Akhir, Jurusan Teknik Sistem Perkapalan – Fakultas Teknologi Kelautan ITS. Surabaya: Indonesia.
- Fajar Prihatama, Erlan. 2011. Rancang Bangun Pengendali Jarak Jauh Arah dan Laju Kendaraan Hover dengan 2 Tenaga Dorong. Proceedings Tugas Akhir, Jurusan Teknik Elektronika – Politeknik Elektronika Negeri Surabaya. Surabaya: Indonesia.
- Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik*. Jakarta: Erlangga.
- Rosdianto, Rio. 2012. Pengendalian Kesetimbangan Per. Axis pada Quardcopter Menggunakan Kontroler Proporsional Integral Deferenensial (PID) Berbasis Mikrokontroler ATmega 168-20AU. Proceedings Tugas Akhir, Jurusan Teknik Elektro – Fakultas Teknik UB. Malang: Indonesia.
- VISHAY. 2011. *Reflective Optical Sensor with Transistor Output*. www.vishay.com.
- Yonathan, Yoneth. 2012. Sistem Pengontrolan Gerak *Hover Tricopter* Berbasis Mikrokontroler. Tugas Akhir, Jurusan Teknik Elektro – Fakultas Teknik Universitas Brawijaya. Malang: Indonesia.

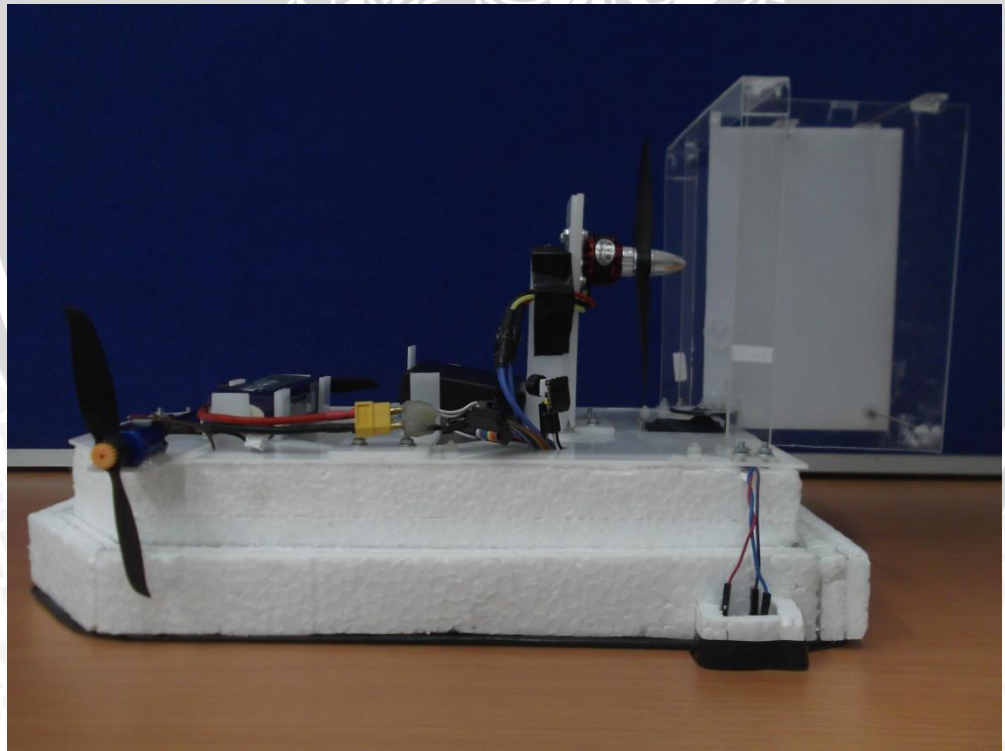
LAMPIRAN 1

Gambar Alat dan Pengujian

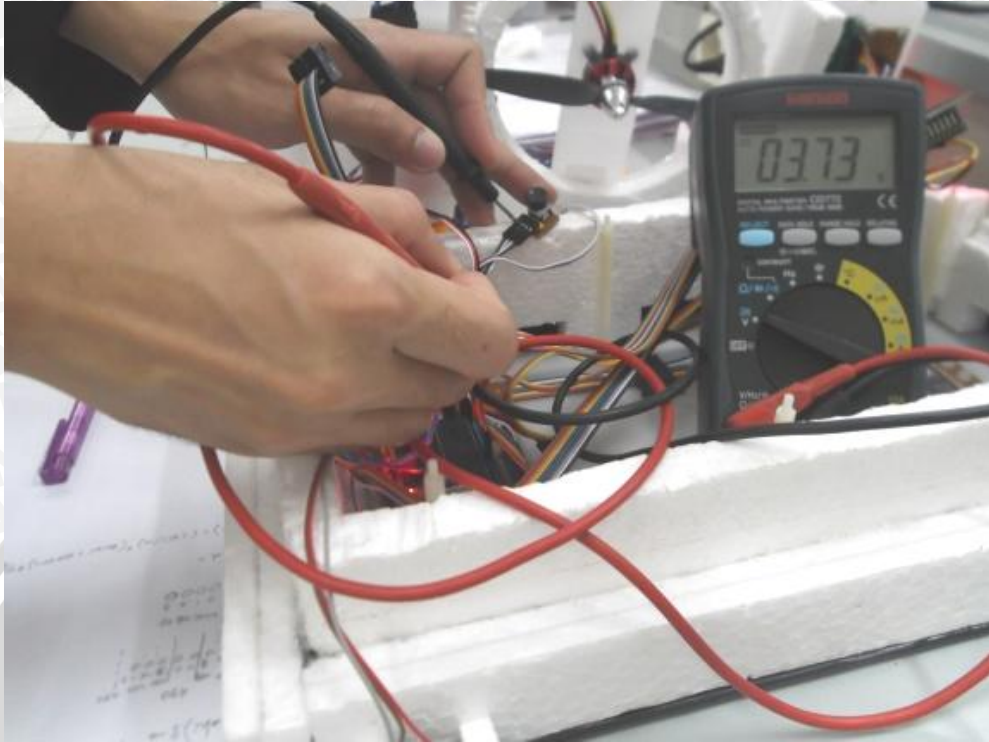




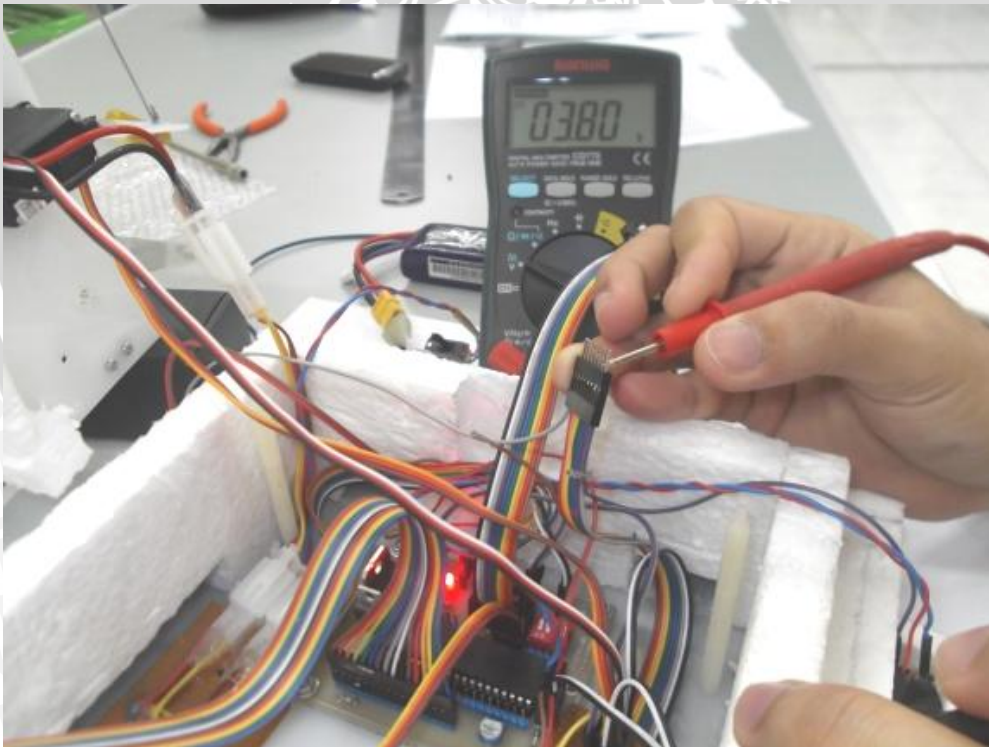
Robot Hovercraft Line Follower Tampak Atas



Robot Hovercraft Line Follower Tampak Samping

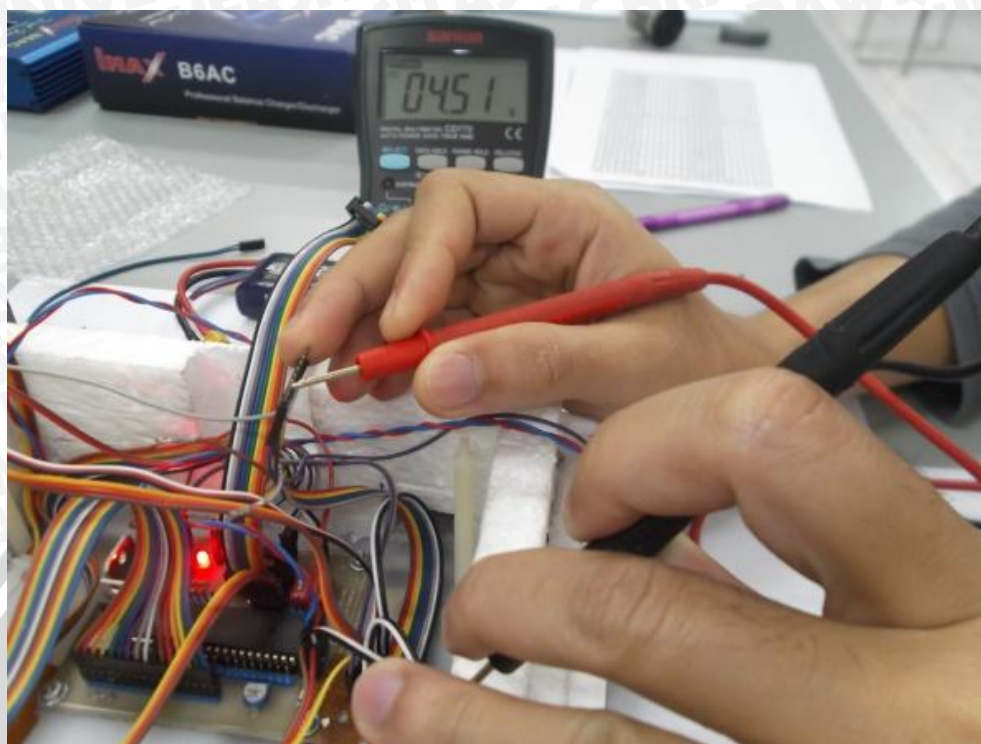


Pengujian Sensor S0 Mengenai Garis Hitam (Pengujian, 2013)

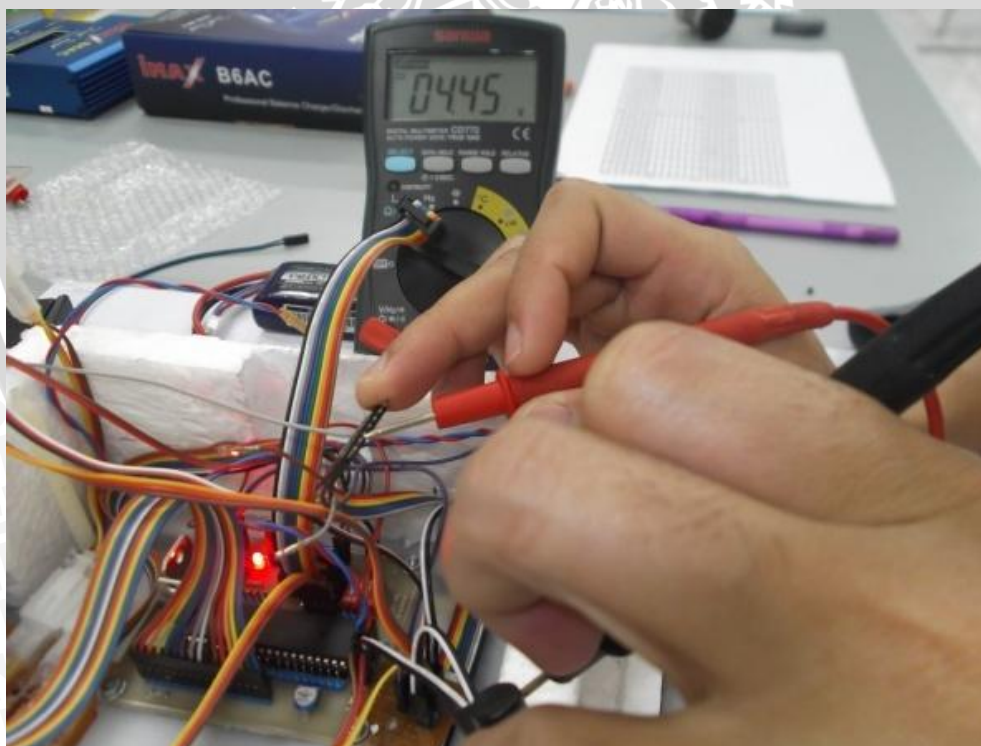


Pengujian Sensor S1 Mengenai Garis Hitam (Pengujian, 2013)

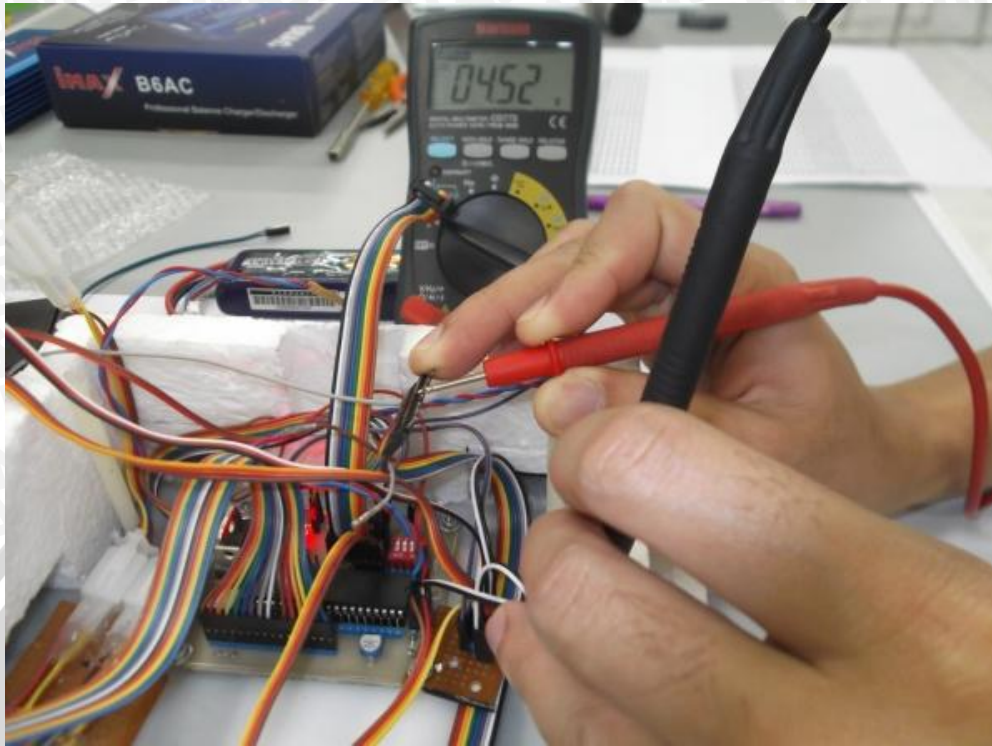




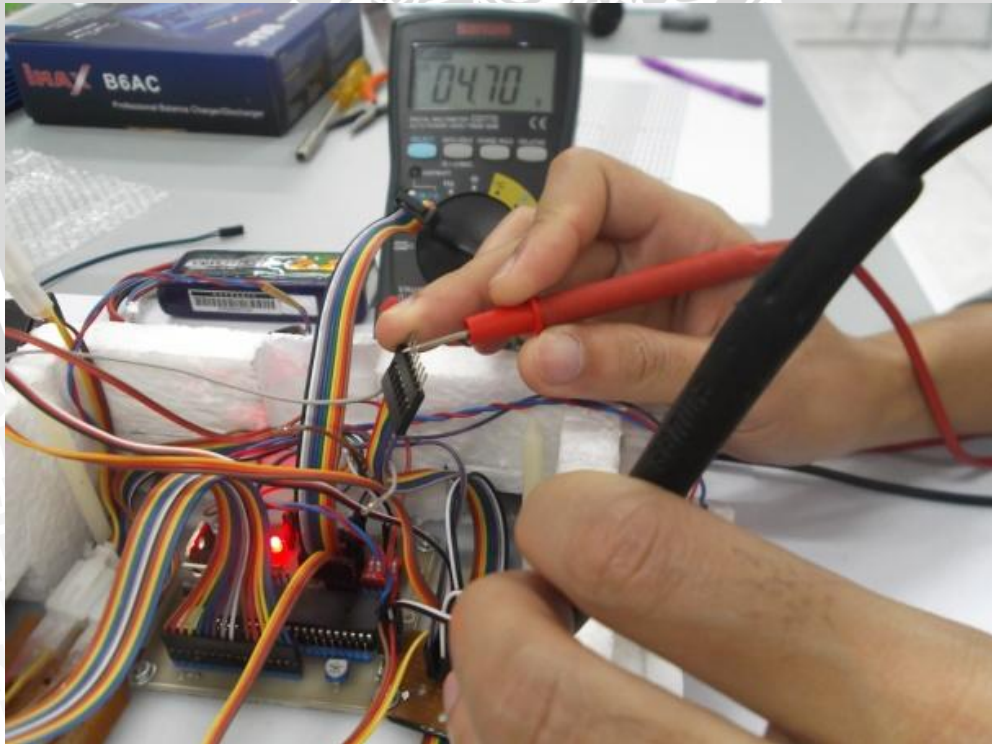
Pengujian Sensor S2 Mengenai Garis Hitam (Pengujian, 2013)



Pengujian Sensor S3 Mengenai Garis Hitam (Pengujian, 2013)

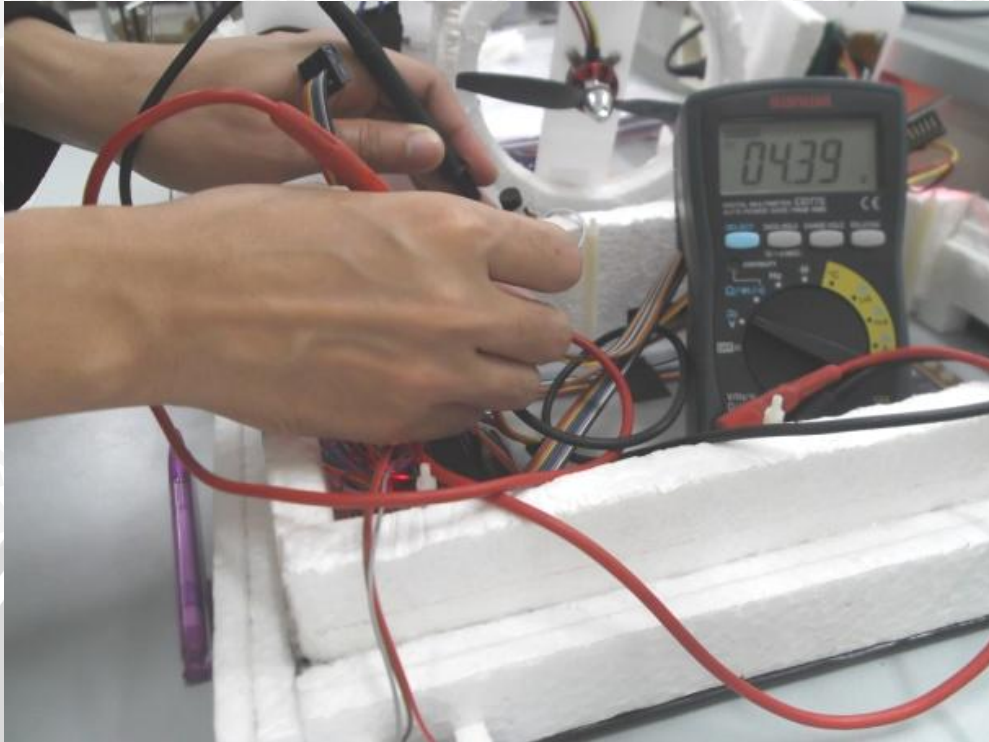


Pengujian Sensor S5 Mengenai Garis Hitam (Pengujian, 2013)

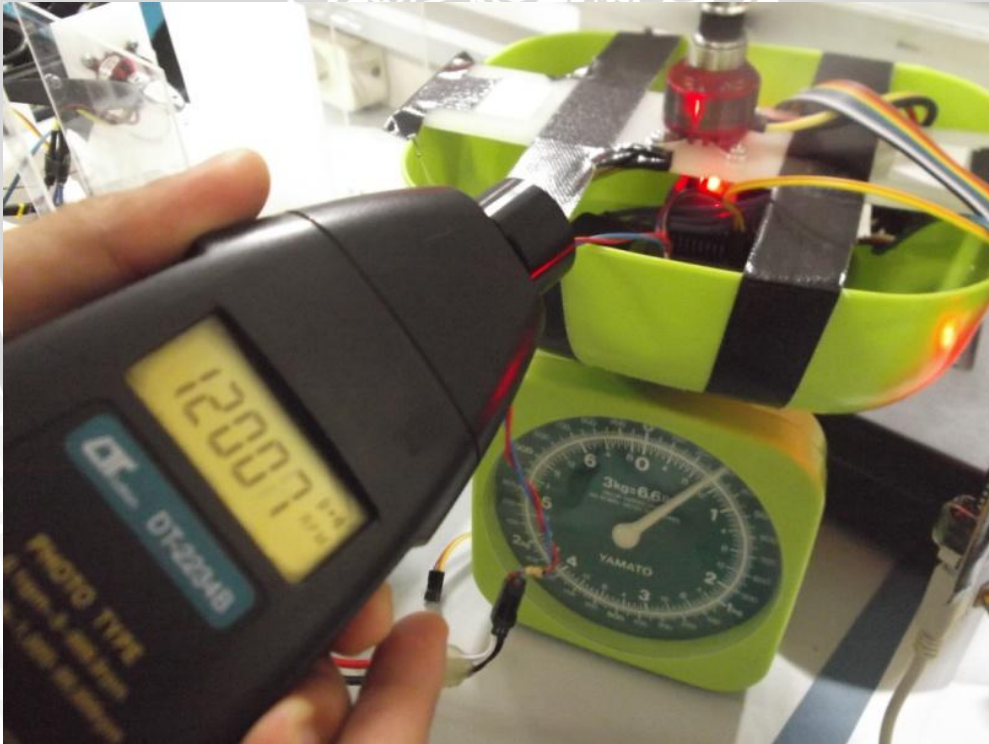


Pengujian Sensor S6 Mengenai Garis Hitam (Pengujian, 2013)



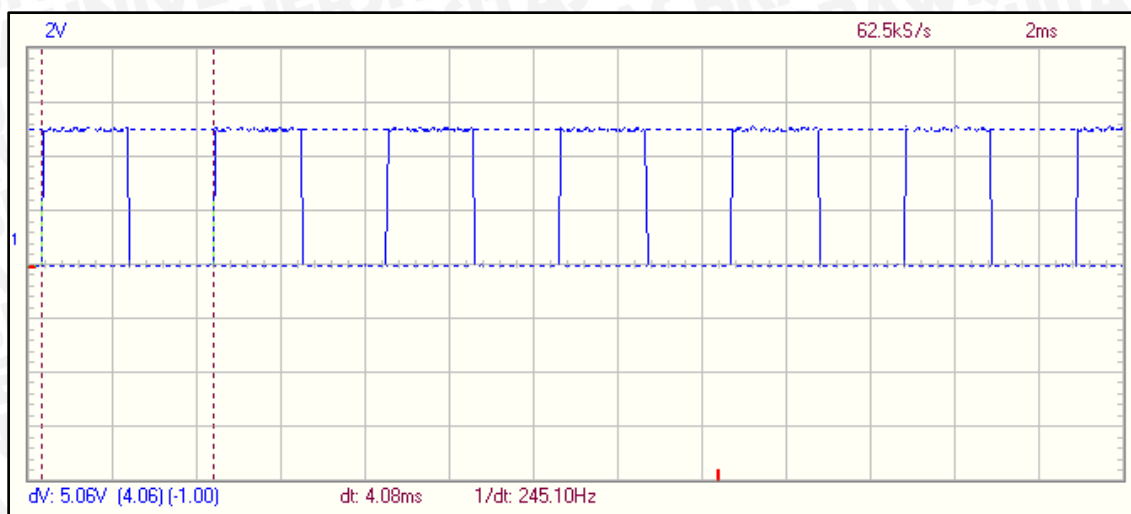


Pengujian Sensor S7 Mengenai Garis Hitam (Pengujian, 2013)



RPM dan Thrust Motor DC Brushless Hover (Pengujian, 2013)

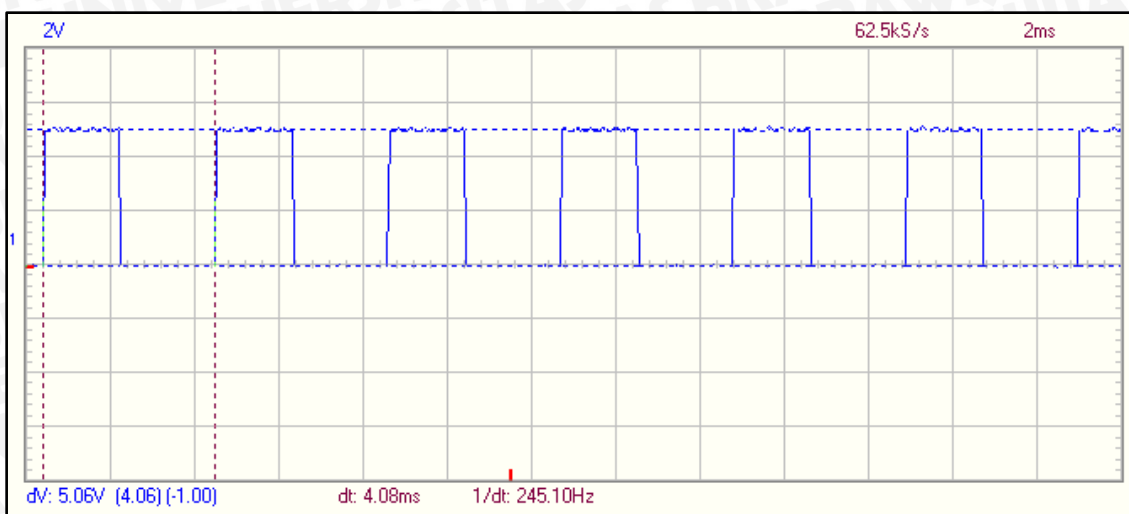




Sinyal PWM Motor Servo pada Sudut 55° (Pengujian, 2013)

Waveform Parameters		CH1	CH2
Amplitude:		<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> DC Mean			
<input type="checkbox"/> Max			
<input type="checkbox"/> Min			
<input type="checkbox"/> Peak-to-Peak			
<input type="checkbox"/> High			
<input type="checkbox"/> Low			
<input checked="" type="checkbox"/> Amplitude	5.06 V		
<input type="checkbox"/> AC RMS			
<input type="checkbox"/> AC dBV			
<input type="checkbox"/> AC dBm			
<input type="checkbox"/> AC+DC RMS			
<input type="checkbox"/> AC+DC dBV			
<input type="checkbox"/> AC+DC dBm			
Timing:			
<input checked="" type="checkbox"/> Duty Cycle	50.4 %		
<input checked="" type="checkbox"/> Positive Width	2.06 ms		
<input checked="" type="checkbox"/> Negative Width	2.03 ms		
<input type="checkbox"/> Rise Time			
<input type="checkbox"/> Fall Time			
<input checked="" type="checkbox"/> Period	4.09 ms		
<input checked="" type="checkbox"/> Frequency	244 Hz		
<input type="checkbox"/> Phase			
<input type="button" value="Select All"/> <input type="button" value="Unselect All"/> <input type="button" value="Options..."/> <input type="button" value="Close"/> <input type="button" value="Help"/>			

Tampilan Parameter PWM Motor Servo pada Sudut 55° (Pengujian, 2013)



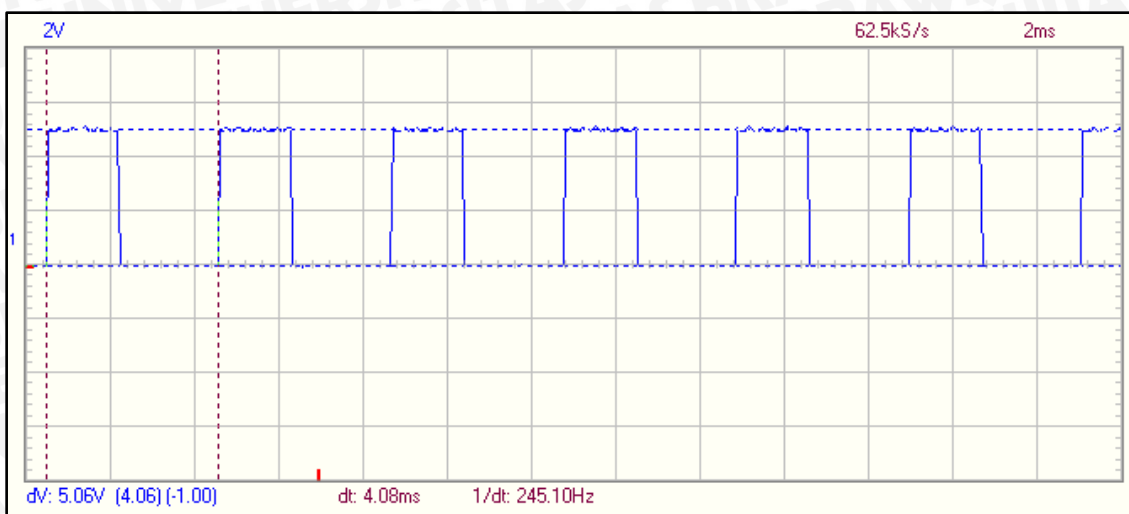
Sinyal PWM Motor Servo pada Sudut 70° (Pengujian, 2013)

Waveform Parameters

Parameter	CH1	CH2
Amplitude:	<input checked="" type="checkbox"/> CH1	<input type="checkbox"/> CH2
<input type="checkbox"/> DC Mean		
<input type="checkbox"/> Max		
<input type="checkbox"/> Min		
<input type="checkbox"/> Peak-to-Peak		
<input type="checkbox"/> High		
<input type="checkbox"/> Low		
<input checked="" type="checkbox"/> Amplitude	5.06 V	
<input type="checkbox"/> AC RMS		
<input type="checkbox"/> AC dBV		
<input type="checkbox"/> AC dBm		
<input type="checkbox"/> AC+DC RMS		
<input type="checkbox"/> AC+DC dBV		
<input type="checkbox"/> AC+DC dBm		
Timing:		
<input checked="" type="checkbox"/> Duty Cycle	44.5 %	
<input checked="" type="checkbox"/> Positive Width	1.82 ms	
<input checked="" type="checkbox"/> Negative Width	2.27 ms	
<input type="checkbox"/> Rise Time		
<input type="checkbox"/> Fall Time		
<input checked="" type="checkbox"/> Period	4.10 ms	
<input checked="" type="checkbox"/> Frequency	244 Hz	
<input type="checkbox"/> Phase		

Select All Unselect All Options... Close Help

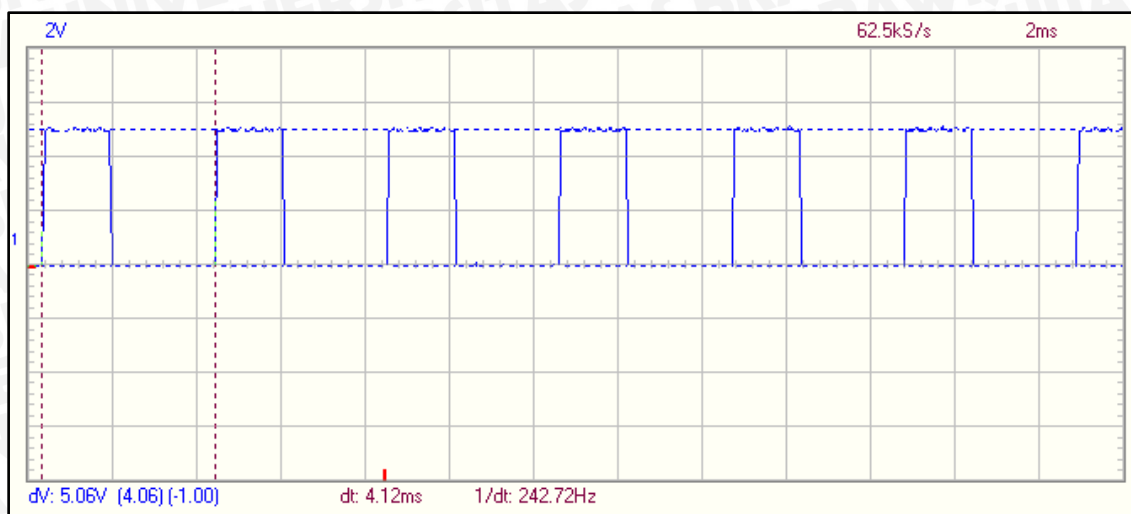
Tampilan Parameter PWM Motor Servo pada Sudut 70° (Pengujian, 2013)



Sinyal PWM Motor Servo pada Sudut 80° (Pengujian, 2013)

Waveform Parameters		
Amplitude:	<input checked="" type="checkbox"/> CH1	<input type="checkbox"/> CH2
<input type="checkbox"/> DC Mean		
<input type="checkbox"/> Max		
<input type="checkbox"/> Min		
<input type="checkbox"/> Peak-to-Peak		
<input type="checkbox"/> High		
<input type="checkbox"/> Low		
<input checked="" type="checkbox"/> Amplitude	5.06 V	
<input type="checkbox"/> AC RMS		
<input type="checkbox"/> AC dBV		
<input type="checkbox"/> AC dBm		
<input type="checkbox"/> AC+DC RMS		
<input type="checkbox"/> AC+DC dBV		
<input type="checkbox"/> AC+DC dBm		
Timing:		
<input checked="" type="checkbox"/> Duty Cycle	41.8 %	
<input checked="" type="checkbox"/> Positive Width	1.71 ms	
<input checked="" type="checkbox"/> Negative Width	2.38 ms	
<input type="checkbox"/> Rise Time		
<input type="checkbox"/> Fall Time		
<input checked="" type="checkbox"/> Period	4.09 ms	
<input checked="" type="checkbox"/> Frequency	244 Hz	
<input type="checkbox"/> Phase		
<input type="button" value="Select All"/> <input type="button" value="Unselect All"/> <input type="button" value="Options..."/> <input type="button" value="Close"/> <input type="button" value="Help"/>		

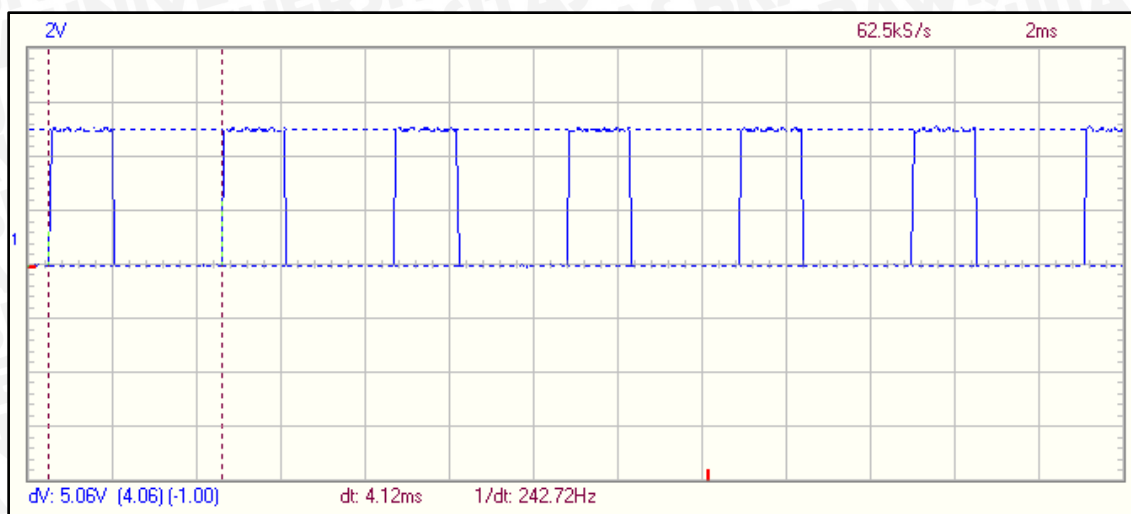
Tampilan Parameter PWM Motor Servo pada Sudut 80° (Pengujian, 2013)



Sinyal PWM Motor Servo pada Sudut 90° (Pengujian, 2013)

Waveform Parameters		CH1	CH2
Amplitude:		<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> DC Mean			
<input type="checkbox"/> Max			
<input type="checkbox"/> Min			
<input type="checkbox"/> Peak-to-Peak			
<input type="checkbox"/> High			
<input type="checkbox"/> Low			
<input checked="" type="checkbox"/> Amplitude	5.06 V		
<input type="checkbox"/> AC RMS			
<input type="checkbox"/> AC dBV			
<input type="checkbox"/> AC dBm			
<input type="checkbox"/> AC+DC RMS			
<input type="checkbox"/> AC+DC dBV			
<input type="checkbox"/> AC+DC dBm			
Timing:			
<input checked="" type="checkbox"/> Duty Cycle	39.1 %		
<input checked="" type="checkbox"/> Positive Width	1.60 ms		
<input checked="" type="checkbox"/> Negative Width	2.49 ms		
<input type="checkbox"/> Rise Time			
<input type="checkbox"/> Fall Time			
<input checked="" type="checkbox"/> Period	4.09 ms		
<input checked="" type="checkbox"/> Frequency	244 Hz		
<input type="checkbox"/> Phase			
<input type="button" value="Select All"/> <input type="button" value="Unselect All"/> <input type="button" value="Options..."/> <input type="button" value="Close"/> <input type="button" value="Help"/>			

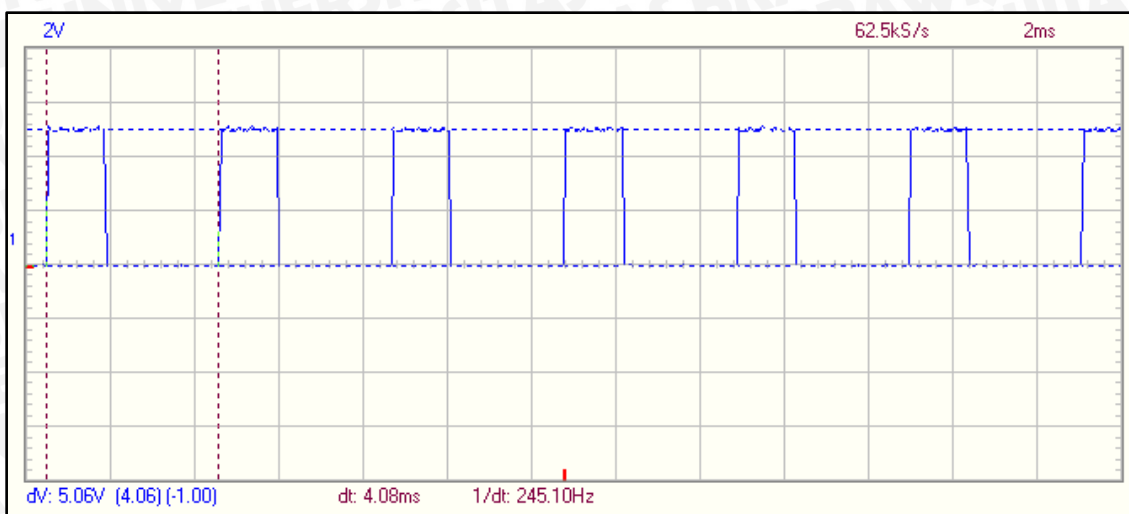
Tampilan Parameter PWM Motor Servo pada Sudut 90° (Pengujian, 2013)



Sinyal PWM Motor Servo pada Sudut 100⁰ (Pengujian, 2013)

Waveform Parameters		CH1	CH2
Amplitude:			
<input type="checkbox"/> DC Mean			
<input type="checkbox"/> Max			
<input type="checkbox"/> Min			
<input type="checkbox"/> Peak-to-Peak			
<input type="checkbox"/> High			
<input type="checkbox"/> Low			
<input checked="" type="checkbox"/> Amplitude	5.06 V		
<input type="checkbox"/> AC RMS			
<input type="checkbox"/> AC dBV			
<input type="checkbox"/> AC dBm			
<input type="checkbox"/> AC+DC RMS			
<input type="checkbox"/> AC+DC dBV			
<input type="checkbox"/> AC+DC dBm			
Timing:			
<input checked="" type="checkbox"/> Duty Cycle	36.3 %		
<input checked="" type="checkbox"/> Positive Width	1.49 ms		
<input checked="" type="checkbox"/> Negative Width	2.61 ms		
<input type="checkbox"/> Rise Time			
<input type="checkbox"/> Fall Time			
<input checked="" type="checkbox"/> Period	4.09 ms		
<input checked="" type="checkbox"/> Frequency	244 Hz		
<input type="checkbox"/> Phase			
<input type="button" value="Select All"/> <input type="button" value="Unselect All"/> <input type="button" value="Options..."/> <input type="button" value="Close"/> <input type="button" value="Help"/>			

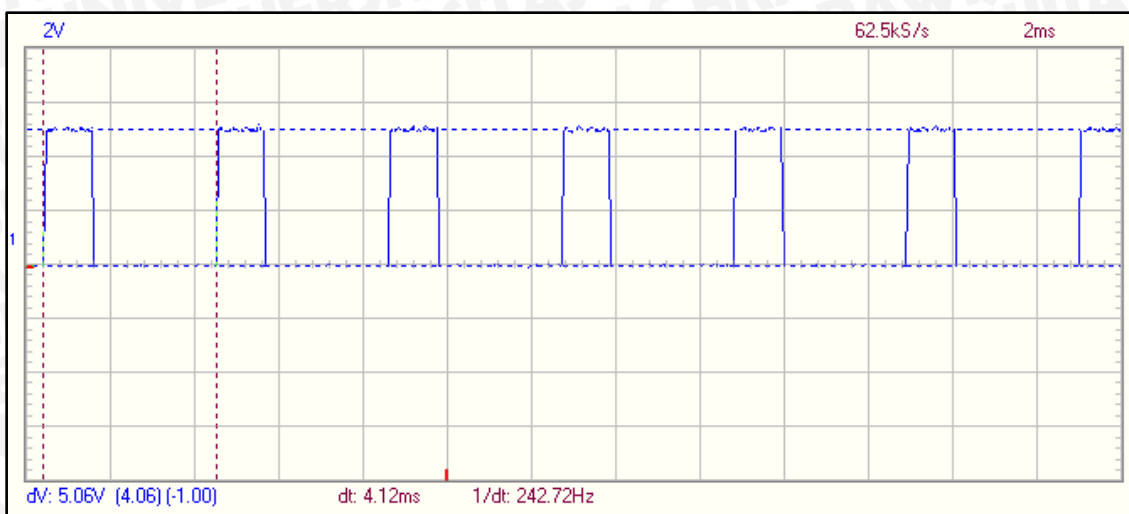
Tampilan Parameter PWM Motor Servo pada Sudut 100⁰ (Pengujian, 2013)



Sinyal PWM Motor Servo pada Sudut 110⁰ (Pengujian, 2013)

Waveform Parameters		
Amplitude:	<input checked="" type="checkbox"/> CH1	<input type="checkbox"/> CH2
<input type="checkbox"/> DC Mean		
<input type="checkbox"/> Max		
<input type="checkbox"/> Min		
<input type="checkbox"/> Peak-to-Peak		
<input type="checkbox"/> High		
<input type="checkbox"/> Low		
<input checked="" type="checkbox"/> Amplitude	5.06 V	
<input type="checkbox"/> AC RMS		
<input type="checkbox"/> AC dBV		
<input type="checkbox"/> AC dBm		
<input type="checkbox"/> AC+DC RMS		
<input type="checkbox"/> AC+DC dBV		
<input type="checkbox"/> AC+DC dBm		
Timing:		
<input checked="" type="checkbox"/> Duty Cycle	33.6 %	
<input checked="" type="checkbox"/> Positive Width	1.38 ms	
<input checked="" type="checkbox"/> Negative Width	2.72 ms	
<input type="checkbox"/> Rise Time		
<input type="checkbox"/> Fall Time		
<input checked="" type="checkbox"/> Period	4.10 ms	
<input checked="" type="checkbox"/> Frequency	244 Hz	
<input type="checkbox"/> Phase		
<input type="button" value="Select All"/> <input type="button" value="Unselect All"/> <input type="button" value="Options..."/> <input type="button" value="Close"/> <input type="button" value="Help"/>		

Tampilan Parameter PWM Motor Servo pada Sudut 110⁰ (Pengujian, 2013)



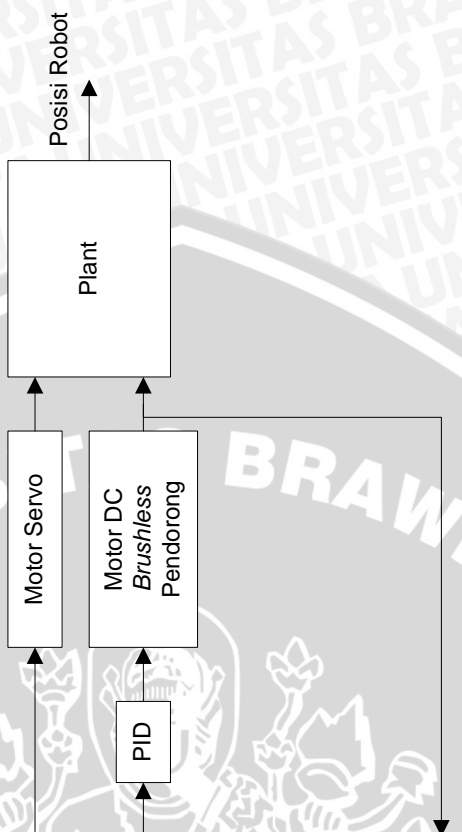
Sinyal PWM Motor Servo pada Sudut 125⁰ (Pengujian, 2013)

Waveform Parameters		
	<input checked="" type="checkbox"/> CH1	<input type="checkbox"/> CH2
Amplitude:		
<input type="checkbox"/> DC Mean		
<input type="checkbox"/> Max		
<input type="checkbox"/> Min		
<input type="checkbox"/> Peak-to-Peak		
<input type="checkbox"/> High		
<input type="checkbox"/> Low		
<input checked="" type="checkbox"/> Amplitude	5.06 V	
<input type="checkbox"/> AC RMS		
<input type="checkbox"/> AC dBV		
<input type="checkbox"/> AC dBm		
<input type="checkbox"/> AC+DC RMS		
<input type="checkbox"/> AC+DC dBV		
<input type="checkbox"/> AC+DC dBm		
Timing:		
<input checked="" type="checkbox"/> Duty Cycle	27.7 %	
<input checked="" type="checkbox"/> Positive Width	1.14 ms	
<input checked="" type="checkbox"/> Negative Width	2.96 ms	
<input type="checkbox"/> Rise Time		
<input type="checkbox"/> Fall Time		
<input checked="" type="checkbox"/> Period	4.10 ms	
<input checked="" type="checkbox"/> Frequency	244 Hz	
<input type="checkbox"/> Phase		
<input type="button" value="Select All"/> <input type="button" value="Unselect All"/> <input type="button" value="Options..."/> <input type="button" value="Close"/> <input type="button" value="Help"/>		

Tampilan Parameter PWM Motor Servo pada Sudut 125⁰ (Pengujian, 2013)

Look Up Table

Logika Sensor	Logika Arah	Logika Penanda	Nilai Sudut Servo	Nilai PWM Motor Pendorong
11100111	lurus	0	0x63	487
11101111	belkir1	0	0x5C	486
11100111	belkir2	0	0x55	485
11100111	belkir3	0	0x46	484
11100111	belkan1	1	0x6A	486
11100111	belkan2	1	0x71	485
11100111	belkan3	1	0x80	484
11100111	belkir1	0	0x5C	486
11100111	belkir2	0	0x55	485
11100111	belkir3	0	0x46	484
11100111	belkan1	0	0x6A	486
11100111	belkan2	0	0x71	485
11100111	belkan3	0	0x80	484
11100111	belkir3	0	0x46	484
11100111	belkan3	1	0x80	484
11100111	belkan3	1	0x80	484
11100111	belkir3	0	0x46	484
11100111	belkan3	1	0x80	484
11100111	belkir3	0	0x46	484
11100111	belkan3	1	0x80	484
11100111	belkir3	0	0x46	484
11100111	belkan3	1	0x80	484
11100111	belkir3	0	0x46	484
11100111	belkan3	0	0x46	484



LAMPIRAN 2

Datasheet Komponen



LAMPIRAN 3

Program Keseluruhan



```

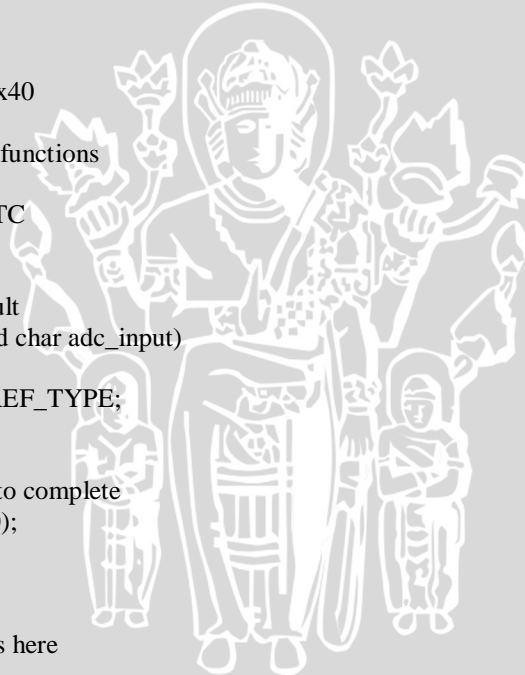
/*****
This program was produced by the
CodeWizardAVR V1.24.8d Professional
Automatic Program Generator
© Copyright 1998-2006 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com
Project : Hovercraft Line Follower
Version :
Date : 6/13/2013
Author : Adeck Aprilyan K
Company : Teknik Elektro, Universitas Brawijaya
Comments:
Chip type : ATmega8535
Program type : Application
Clock frequency : 16.000000 MHz
Memory model : Small
External SRAM size : 0
Data Stack size : 128
*****/

```

```

#include <mega8535.h>
#include <delay.h>
#include <stdio.h>
#define ADC_VREF_TYPE 0x40
#define PWM OCR1B
// Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input|ADC_VREF_TYPE;
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}
// Declare your global variables here
void lurus();
void belkan1();
void belkan2();
void belkan3();
void belkir1();
void belkir2();
void belkir3();
//Tombol Yang Dipakai
#define sw_ok PINB.3
#define sw_down PINB.2
#define sw_up PINB.1
#define sw_cancel PINB.0
//int cek;
//Variabel di eeprom
eeprom int Kp=0;
eeprom float Ki=0;

```



```

eeprom float Kd=0;
eeprom int Ts=0;
eeprom int Kecepatan=0;
eeprom int Set_point=0;
//Nilai-nilai error
eeprom int e0=0;
eeprom int e1=0;
eeprom int e2=0;
eeprom int e3=0;
eeprom int e4=0;
eeprom int e5=0;
eeprom int e6=0;
eeprom int e7=0;
int a,b,c,d,e,f,sampingkir,sampingkan,penanda=0;
int error,error1=0,nilai;
float nil_pid=0;
char lcd_buff[33];
void adeck()
{
  lcd_gotoxy(0,0);
  lcd_putsf("Present");
  delay_ms(1000);
  lcd_clear();
  lcd_gotoxy(0,0);
  lcd_putsf("Adeck A. K.");
  delay_ms(500);
  lcd_gotoxy(0,1);
  lcd_putsf("0910630021");
  delay_ms(1000);
  lcd_gotoxy(0,0);
  lcd_putsf("Hovercraft");
  delay_ms(500);
  lcd_gotoxy(0,1);
  lcd_putsf("Line Follower");
  delay_ms(500);
  lcd_clear();
}
void setting_awal()
{
  PORTB.0=1;
  PORTB.1=1;
  PORTB.2=1;
  PORTB.3=1;
  set:
  Kp=Kp;
  //delay_ms(100);
  if(sw_up==0){Kp++;delay_ms(10);}
  if(sw_down==0){Kp=Kp-1;delay_ms(10);}
  if(Kp>25){Kp=0;delay_ms(10);}
  if(Kp<0){Kp=25;delay_ms(10);}
  lcd_clear();
  lcd_gotoxy(0,0);
  lcd_putsf("*****");
  lcd_gotoxy(0,1);
  sprintf(lcd_buff,"Nilai Kp:%i",Kp);
  lcd_puts(lcd_buff);
  delay_ms(100);

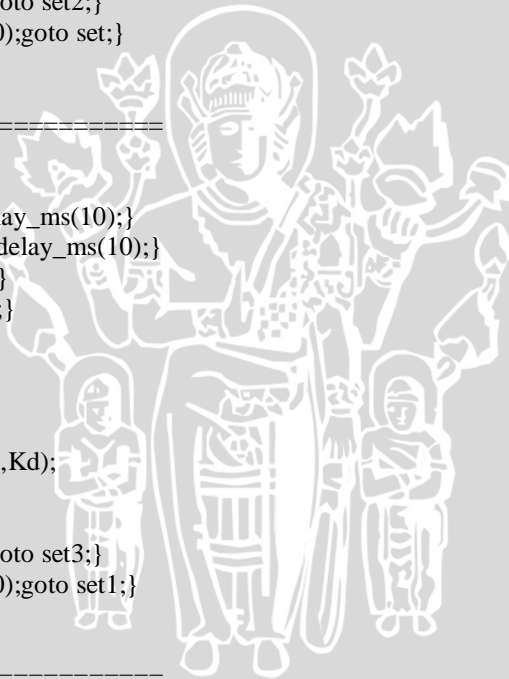
```



```

if(sw_ok==0)
{
delay_ms(150);goto set1;
}
else goto set;
//=====
set1:
Ki=Ki;
if(sw_up==0) {Ki=Ki+0.033;delay_ms(10);}
if(sw_down==0) {Ki=Ki-0.033;delay_ms(10);}
if(Ki>10){Ki=0;delay_ms(10);}
if(Ki<0) {Ki=10;delay_ms(10);}
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("*****");
lcd_gotoxy(0,1);
sprintf(lcd_buff,"Nilai Ki:%.3f",Ki);
lcd_puts(lcd_buff);
delay_ms(100);
if(sw_ok==0) {delay_ms(150);goto set2;}
if(sw_cancel==0){delay_ms(150);goto set;}
else
goto set1;
//=====
set2:
Kd=Kd;
if(sw_up==0) {Kd=Kd+0.18;delay_ms(10);}
if(sw_down==0) {Kd=Kd-0.18;delay_ms(10);}
if(Kd>10){Kd=0;delay_ms(10);}
if(Kd<0) {Kd=10;delay_ms(10);}
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("*****");
lcd_gotoxy(0,1);
sprintf(lcd_buff,"Nilai Kd:%.2f",Kd);
lcd_puts(lcd_buff);
delay_ms(100);
if(sw_ok==0) {delay_ms(150);goto set3;}
if(sw_cancel==0){delay_ms(150);goto set1;}
else
goto set2;
//=====
set3:
Ts=10;
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Time Sampling");
lcd_gotoxy(0,1);
sprintf(lcd_buff,"Delay:10ms");
lcd_puts(lcd_buff);
delay_ms(100);
if(sw_ok==0) {delay_ms(150);goto set4;}
if(sw_cancel==0){delay_ms(150);goto set2;}
else
goto set3;
//=====
set4:

```




```

{error=Set_point-e2;}
else if ((c==0)&&(d==0)&&(e==0))
{error=Set_point-e3;}
else if ((d==0)&&(e==0)&&(sampingkir==0))
{error=Set_point-e1;}
else if ((e==0)&&(f==0)&&(sampingkir==0))
{error=Set_point-e2;}
else if ((f==0)&&(sampingkir==0))
{error=Set_point-e3;}
else if ((b==0)&&(c==0)&&(sampingkan==0))
{error=Set_point-e1;}
else if ((a==0)&&(b==0)&&(sampingkan==0))
{error=Set_point-e2;}
else if ((a==0)&&(sampingkan==0))
{error=Set_point-e3;}
else if (a==0||b==0)
{error=Set_point-e3;}
else if (e==0||f==0)
{error=Set_point-e3;}
}
else if((sampingkan==0)&&(sampingkir==1))
{if(penanda==1)
{error=Set_point-e3;}
else if(penanda==0)
{error=Set_point-e3;}
}
else if((sampingkan==1)&&(sampingkir==0))
{if(penanda==1)
{error=Set_point-e3;}
else if(penanda==0)
{error=Set_point-e3;}
}
else if((sampingkan==0)&&(sampingkir==0))
{if(penanda==1)
{error=Set_point-e3;}
else if(penanda==0)
{error=Set_point-e3;}
}
}
void starter()
{
PORTB.6=0;
PORTB.7=0;
OCR1B = 175;
OCR1A = 175;
delay_ms(8000);
OCR1B = 484;
OCR1A = 425;
delay_ms(13000);
}
// Timer2 output compare interrupt service routine
interrupt [TIM2_COMP] void timer2_comp_isr(void)
{
// Place your code here
TCNT2 = 0x00; //membuat nilai inisial 0
TCCR2 = 0x00; //membuat OCR2 tidak aktif skala 0
// ADCSRA.7 = 1; //ADC enable aktif saat 1

```



```
//ADCSRA.6 = 1; //memulai pengambilan data
// timerku=timerku+1;
TCCR2 = 0x0A; //membuat frekuensi timer 8 kHz
//pengujian penyamplingan
// PORTB.0=~PORTB.0;
//
}
// ADC interrupt service routine
interrupt [ADC_INT] void adc_isr(void)
{
ADCSRA.6 = 0; //mengakhiri pengamban data
ADCSRA.7 = 0; //ADC enable tidak aktif saat 0
//memberikan nilai kuantisasi sesuai output kuantisasi ADC
}
void main(void)
{
// Declare your local variables here
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=P State6=P State5=T State4=T State3=P State2=P State1=P State0=P
PORTB=0x00;
DDRB=0b11001111;
// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;
// Port D initialization
// Func7=Out Func6=In Func5=Out Func4=Out Func3=In Func2=In Func1=In Func0=In
// State7=0 State6=T State5=0 State4=0 State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0xB0;
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 250.000 kHz
// Mode: Fast PWM top=03FFh
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
```

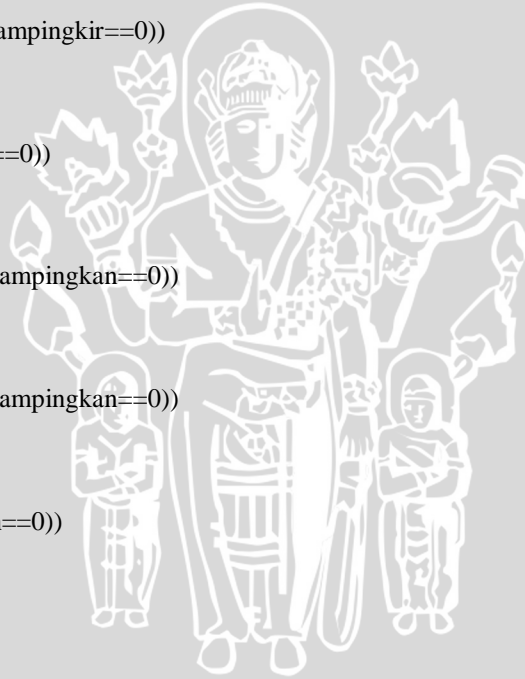
```
// Compare B Match Interrupt: Off
TCCR1A=0xA3;
TCCR1B=0x0B;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: 250.000 kHz
// Mode: Fast PWM top=FFh
// OC2 output: Non-Inverted PWM
ASSR=0x00;
TCCR2=0x6E;
TCNT2=0x00;
OCR2=0x63;
// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x80; //bit 7 aktif maksudnya timer2 yang digunakan
ACSR=0x80; // bit 7 bernilai 1 >> mengaktifkan fungsi analog komparator
SFIOR=0x00;
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x67;
// ADC initialization
// ADC Clock frequency: 1000.000 kHz
// ADC Voltage Reference: AVCC pin
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=0b01000000; // bit 7 dan 6 bernilai 01 berarti menggunakan AVCC
ADCSRA=0b11001100; // bit 7 dan 6 untuk memulai dan mengakhiri ADC ,faktor pembagi 16, bit
3 mengaktifkan interrupt ADC
// SPI initialization
// SPI disabled
SPCR=0x00;
// TWI initialization
// TWI disabled
TWCR=0x00;
// LCD module initialization
lcd_init(16);
```

```
lcd_clear();
adeck();
setting_awal();
starter();
while (1)
{
int data0,data1,data2,data3,data4,data5,data6,data7;
int a,b,c,d,e,f,sampingkir,sampingkan,penanda=0;
// Place your code here
while(1)
{
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("PWM : ");
sprintf(lcd_buff,"%i",OCR1B);
lcd_puts(lcd_buff);
lcd_gotoxy(0,1);
lcd_putsf("ERROR : ");
sprintf(lcd_buff,"%i",error);
lcd_puts(lcd_buff);
data0=read_adc(0);
data1=read_adc(1);
data2=read_adc(2);
data3=read_adc(3);
data4=read_adc(4);
data5=read_adc(5);
data6=read_adc(6);
data7=read_adc(7);
if(data0>=340) sampingkir=0; else sampingkir=1;
if(data1>=340) a=0; else a=1;
if(data2>=340) b=0; else b=1;
if(data3>=340) c=0; else c=1;
if(data4>=340) d=0; else d=1;
if(data5>=340) e=0; else e=1;
if(data6>=340) f=0; else f=1;
if(data7>=340) sampingkan=0; else sampingkan=1;
PID();
nil_pid=((Kp*(-error))+(Ki*((-error)+error1)*Ts)+((Kd/Ts)*((-error)-error1));
error1=(-error);
if((nil_pid>0)&&(nil_pid<10)) {nilai=0;}
else if((nil_pid>50)&&(nil_pid<100)) {nilai=-1;}
else if((nil_pid>150)&&(nil_pid<235)) {nilai=-2;}
else if((nil_pid>240)&&(nil_pid<400)) {nilai=-3;}
if((sampingkan==1)&&(sampingkir==1))
{if ((c==0)&&(d==0))
{
lurus();
delay_ms(10);
}
else if ((c==0))
{belkir1();
delay_ms(10);
penanda=0;}
else if ((c==0)&&(b==0)||((b==0)))
{belkir2();
delay_ms(10);
penanda=0;}
}
```

```

else if ((d==0)&&(c==0)&&(b==0))
{belkir3();
delay_ms(10);
penanda=0;}
else if ((d==0))
{belkan1();
delay_ms(10);
penanda=1;}
else if ((d==0)&&(e==0)||((e==0)))
{belkan2();
delay_ms(10);
penanda=1;}
else if ((c==0)&&(d==0)&&(e==0))
{belkan3();
delay_ms(10);
penanda=1;}
else if ((d==0)&&(e==0)&&(sampingkir==0))
{belkir1();
delay_ms(10);
penanda=0;}
else if ((e==0)&&(f==0)&&(sampingkir==0))
{belkir2();
delay_ms(10);
penanda=0;}
else if ((f==0)&&(sampingkir==0))
{belkir3();
delay_ms(10);
penanda=0;}
else if ((b==0)&&(c==0)&&(sampingkan==0))
{belkan1();
delay_ms(10);
penanda=0;}
else if ((a==0)&&(b==0)&&(sampingkan==0))
{belkan2();
delay_ms(10);
penanda=0;}
else if ((a==0)&&(sampingkan==0))
{belkan3();
delay_ms(10);
penanda=0;}
else if (a==0||b==0)
{belkir3();
delay_ms(10);
penanda=0;}
else if (e==0||f==0)
{belkan3();
delay_ms(10);
penanda=1;}
}
else if((sampingkan==0)&&(sampingkir==1))
{if(penanda==1)
{belkan3();
delay_ms(10);}
else if(penanda==0)
{belkir3();
delay_ms(10);}
}

```



```

else if((sampingkan==1)&&(sampingkir==0))
{if(penanda==1)
{belkan3();
delay_ms(10);}
else if(penanda==0)
{belkir3();
delay_ms(10);}
}
else if((sampingkan==0)&&(sampingkir==0))
{if(penanda==1)
{belkan3();
delay_ms(10);}
else if(penanda==0)
{belkir3();
delay_ms(10);}
}
};
}
void lurus()
{OCR2=0x63;
OCR1B=Kecepatan+(nilai);
error=Set_point-e0;
PORTB.6=0;
PORTB.7=0;} //90 derajat
void belkan1()
{OCR2=0x6A;
OCR1B=Kecepatan+(nilai);
error=Set_point-e1;
PORTB.6=0;
PORTB.7=1;} //80 derajat
void belkan2()
{OCR2=0x71;
OCR1B=Kecepatan+(nilai);
error=Set_point-e2;
PORTB.6=0;
PORTB.7=1;} //70 derajat
void belkan3()
{OCR2=0x80;
OCR1B=Kecepatan+(nilai);
error=Set_point-e3;
PORTB.6=0;
PORTB.7=1;} //55 derajat
void belkir1()
{OCR2=0x5C;
OCR1B=Kecepatan+(nilai);
error=Set_point-e1;
PORTB.6=1;
PORTB.7=0;} //100 derajat
void belkir2()
{OCR2=0x55;
OCR1B=Kecepatan+(nilai);
error=Set_point-e2;
PORTB.6=1;
PORTB.7=0;} //110 derajat
void belkir3()
{OCR2=0x46;

```



OCR1B=Kecepatan+(nilai);
error=Set_point-e3;
PORTB.6=1;
PORTB.7=0;} //125 derajat

