

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT, karena dengan rahmat, taufik dan hidayah-Nya skripsi ini dapat diselesaikan. Skripsi berjudul **“Perancangan Sistem Kontrol Pengereman Motor DC Pada Mobil Listrik”** 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:

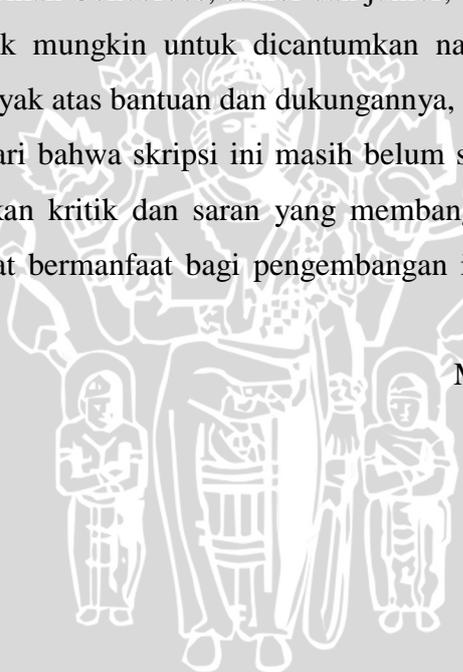
1. Allah SWT atas rahmat dan hidayah yang diberikan dan Rasulullah Muhammad SAW semoga sholawat dan salam tetap tercurah kepada beliau,
2. Ayah Siswandi. dan Ibu Mistiari atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, dukungan moral dan material, serta telah banyak mendoakan kelancaran penulis hingga terselesaikan nya skripsi ini,
3. Luluk ifa dan Masulfa istiqomah, adik dan kakak yang selalu menemani keseharian dan memberi keceriaan,
4. Nenek, Tante Eni Sekeluarga, Cak ipul Sekeluarga dan Budhe Nanik sekeluarga, terimakasih atas tumpangan tempat tinggal dan supportnya selama ini,
5. Arief f, Q-co, Bojes, Agus, Anas, Timboel, Arsyad, Limbad, Maul, Okta, faiz, Andika, Saprol, Bakwan dkk, terima kasih atas dukungan dan motivasinya teman seperjuangan,
6. Semua keluarga besar dan saudara yang sudah mendukung,
7. Bapak M. Aziz Muslim, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya dan Bapak Hadi Suyono, S.T., M.T., Ph.D. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya,
8. Bapak Goegoes Dwi Nusantoro ST., MT. selaku Dosen Pembimbing I atas segala ilmu, bimbingan, nasehat, gagasan, ide, saran, motivasi dan bantuan yang telah diberikan,

9. Bapak Ir. Moch.Rusli, Dipl.-Ing. selaku Dosen Pembimbing II atas segala ilmu, bimbingan, nasehat, gagasan, ide, saran, motivasi dan bantuan yang telah diberikan,
10. Bapak Ali Mustofa, S.T., M.T. selaku Dosen Pembimbing akademik penulis atas segala nasehat dan bimbingan yang telah diberikan,
11. Sahabat Nakojop (Nyok, Pribadhi, Blontang, Tiong, Dodid, Mifta, Dakar, Wiyogo, Umar, Aldy, Wanyik, Arnas, Ceper, Fandy, MONO, Jimbon, Hoyi, Samid, Ape, Ade, Pepi, Rio), Parkiran (Cak Muktar, Pak Lan, Bang Ivan, Rendi, Irwan, Engga, Dita, Candra, Ulin), CL (Praja, Bambi, Smigol, Nggurah, Akbar, Gopar, Cino, Supir, Kipot, Daus),Terima kasih terutama pada Agustri, Anas, Arnas atas bantuannya menyelesaikan skripsi ini,
12. Seluruh teman-teman Concordes, senior dan junior, asisten lab 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, Agustus 2015

Penulis



ABSTRAK

Fajar Lutfialayubi, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, 2015, *Perancangan Sistem Kontrol Pengereman Motor DC secara Eleltris pada Mobil Listrik*, Dosen Pembimbing: Goegoes Dwi Nusantoro, ST., MT., dan Ir. Moch. Rusli, Dipl.-Ing.

Skripsi ini membahas tentang perancangan Sistem Pengereman pada Motor DC secara elektrik dengan jarak kurang dari 300cm dari penghalang dengan menggunakan Arduino UNO menggunakan parameter dari *controller Proporsional Integral Deferenisial (PID)*. Dengan menggunakan sistem *Pulse Width Modulation (PWM)* maka dapat dilakukan pemberhentian dengan cara memberikan pengereman secara otomatis pada motor DC sebagai penggerak. Besaran harga keluaran pada algoritma PID ini yang dijadikan acuan dalam pembangkit PWM, sehingga terjadi pengereman kecepatan motor DC yang dapat berhenti tepat pada jarak 50cm dari penghalang. Sensor yang di gunakan sebagai pengukur jarak adalah sensor ultrasonik.

Pemograman dilakukan menggunakan software Arduino 1.0.6. Pada skripsi ini dicari harga kombinasi dari Variabel PID yaitu K_p, K_i, K_d dengan metode Hand Tunning. Hasil parameter kontroler PID diperoleh nilai $K_p=0.02$, $K_i=0$, $K_d=0.054$. dengan di dapatkannya nilai K_p , K_i , dan K_d tersebut, sistem dapat berjalan dengan baik karena ketika sensor ping bekerja dengan jarak kurang dari 300cm motor DC akan melakukan pengereman dan akan berhenti langsung ketika kurang dari 50cm.

Kata Kunci : *Sensor Ultrasonik, (PING))), motor DC, Motor DC, Hand Tunning, Kontrol,*

BAB I PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi masa kini berkembang sangat pesat. Hal ini dapat dibuktikan dengan banyaknya inovasi-inovasi yang telah di ciptakan di dunia. Salah satu perkembangan teknologi saat ini adalah bermunculannya mobil listrik. Mobil listrik merupakan teknologi terbaru pada kendaraan roda empat yang memanfaatkan sumber listrik sebagai bahan bakarnya dan motor listrik sebagai penggerakannya. Pada perkembangan penciptaan mobil listrik, kecelakaan merupakan suatu kejadian saat pengemudi mengalami kelalaian dalam berkendara atau kegagalan pada sistem pengereman manual. Tingginya angka kecelakaan lalu lintas menjadi salah satu momok bagi para pengemudi kendaraan bermotor. Untuk menghindari hal itu, etika berkendara memang perlu. Namun, kualitas sistem keamanan kendaraan juga sangat mempengaruhi keselamatan pengemudi.

Dengan perkembangan ilmu pengetahuan dan teknologi yang telah ada memungkinkan manusia untuk membuat sistem keamanan pada kendaraan bermotor. Kecelakaan sering terjadi karena pengemudi tidak dapat mengendalikan laju kendaraan dengan segera ketika secara mendadak ada obyek di depan. Terlebih lagi ini terjadi saat kendaraan melaju dengan kecepatan tinggi. Hal ini dapat diatasi dengan membuat sistem pengereman otomatis. Laju kendaraan akan melambat meskipun pengemudi tidak menarik tuas rem.

Berdasarkan permasalahan tersebut maka perlu dibuat suatu sistem pengaman otomatis pada motor listrik dengan cara pengurangan dan penambahan sumber tegangan menggunakan metode PWM (*pulse width modulation*) serta menggunakan sensor ultrasonik sebagai indikator putaran motor DC pada mobil listrik yang mampu memberikan pembacaan adanya benda pantul yang menghalangi laju mobil listrik sehingga laju mobil listrik akan berhenti sebelum terjadi tabrakan. Gelombang ultrasonik adalah gelombang dengan besar frekuensi di atas frekuensi gelombang suara yaitu lebih dari 20 KHz. Sensor ultrasonik digunakan sebagai detektor jarak antara kendaraan dengan obyek di depannya.

Pada prinsipnya, sensor ultrasonik terdiri dari rangkaian pemancar ultrasonik yang disebut transmitter dan rangkaian penerima ultrasonik yang disebut receiver. Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari transmitter ultrasonik. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh receiver ultrasonik. Mikrokontroler digunakan sebagai pengontrol kerja sistem. Mikrokontroler ini akan mengolah data kecepatan rambat gelombang dikali setengah waktu tempuh sehingga diperoleh jarak. Jarak antara kendaraan dan obyek ditampilkan pada LCD (*Liquid Crystal Display*). Pada jarak tertentu, mikrokontroler akan menghentikan putaran motor DC.

1.2 Rumusan Masalah

Berdasarkan latar belakang, dapat disusun rumusan masalah sebagai berikut:

1. Bagaimana perancangan dan metode sistem pengereman otomatis pada mobil listrik?
2. Bagaimana Respon dari kinerja pengereman otomatis dari sensor jarak?

1.3 Batasan Masalah

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan alat akan diberi batasan sebagai berikut:

1. Sistem Pengereman yang di buat merupakan pemodelan atau *prototype*
2. Studi Pembahasan di tekankan pada proses pengereman otomatis berdasar dari sensor jarak/ultrasonik yang mendapat gangguan dari benda padat.
3. Kontroler yang di gunakan menggunakan Arduino UNO
4. Benda padat yang di gunakan untuk riset ini berupa papan kayu dan besi.
5. Kinerja *driver* dan elektronika sistem tidak dibahas mendalam.
6. Parameter gangguan kecelakaan lain seperti medan berlubang dsb diabaikan.

1.4 TUJUAN

Perancangan ini bertujuan untuk membuat sistem keamanan mobil listrik pada pengereman, dimana terjadi secara otomatis saat sensor mendeteksi di bawah jarak aman yang telah di tentukan.

1.5 Sistematika Pembahasan

Skripsi ini terdiri dari enam bab dengan sistematika pembahsan sebagai berikut:

BAB I Pendahuluan

Membahas latar belakang, rumusan masalah, batasan masalah, tujuan dan sitematika pembahasan.

BAB II Tinjauan Pustaka

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan alat .

BAB III Metodologi Penulisan

Membahas metode penelitian dan perencanaan alat.

BAB IV Perancangan dan Pembuatan Alat

Membahas perancangan kontroler Proporsional Integral Deferenensial dan menerapkan ke dalam *software* dikomputer, sehingga sistem dapat bekerja baik.

BAB V Pengujian dan Analisis

Membahas hasil pengujian sistem yang sudah dibuat, serta analisis hasil yang diperoleh.

BAB VI Kesimpulan dan Saran

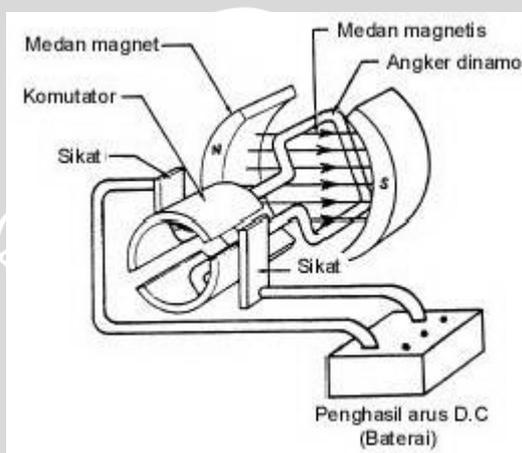
Membahas kesimpulan perancangan ini dan saran-saran yang diperlukan untuk pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 MOTOR DC

Motor listrik merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Motor DC atau sering disebut motor arus searah lebih sering digunakan untuk keperluan yang membutuhkan pengaturan kecepatan dibandingkan dengan motor ac. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen. Motor DC merupakan jenis motor yang menggunakan tegangan searah sebagai sumber tenaganya.



Gambar 2.1 Motor DC sederhana
(sumber: elektronika-dasar.web.id,2015)

Mekanisme kerja untuk motor DC pada pada Gambar 2.1 adalah sebagai berikut:

- 1) Arus listrik dalam medan magnet pada angker dinamo akan memberikan gaya.
- 2) Jika kawat yang membawa arus dibengkokkan menjadi sebuah lingkaran/loop, maka kedua sisi loop, yaitu pada sudut kanan medan magnet akan mendapatkan gaya pada arah yang berlawanan.
- 3) Pasangan gaya pada angker dan magnet akan menghasilkan tenaga putar/torque untuk memutar kumparan.
- 4) Motor memiliki beberapa loop pada dinamonya untuk memberikan tenaga

putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan.

Ada tiga metode sistem pengereman secara elektrik yang dapat diaplikasikan pada motor DC. Berikut tiga metode sistem pengereman pada motor listrik :

1) Pengereman secara *Regenerative*

Pengereman secara *regenerative* adalah pengereman yang dilakukan dengan menghubungkan antara polaritas positif dan negative, sehingga akan didapat putaran motor akan berhenti dengan cepat.

2) Pengereman secara Dinamis

Pengereman ini dilakukan dengan cara memutuskan dua buah terminal fasa motor dari jala-jala dan lalu dihubungkan dengan sumber DC tegangan rendah. Biasanya sumber DC diperoleh dari suatu semi konduktor *rectifier* yang terhubung dengan jala-jala melalui suatu transformator.

Arus DC yang dihubungkan pada kumparan stator akan menghasilkan suatu medan statis. Maka dalam rotor akan diinduksikan suatu gaya gerak listrik E, arus induksi I yang berbanding lurus dengan putaran n.

Torsi yang dihasilkan sebanding dengan arus I dan fluks ϕ .

$$T \approx I\phi \approx n \text{ maka } T \approx n \dots\dots\dots (2.1)$$

3) Pengereman secara *Plugging*

Pengereman yang dilakukan mampu menghentikan motor lebih cepat dengan menggunakan metode yang disebut metode plugging. Prinsip kerjanya adalah membalikkan arus angker dengan cara membalik terminal sumber. Sehingga akan didapat kondisi motor berputar balik seiring polaritas sumber terbalik. Akan tetapi harus diperhatikan bahwa untuk maksud pengereman, maka pada saat arah putaran medan magnetik motor mendadak terbalik, motor diputus dari jala-jala sebelum motor dapat mulai bekerja pada arah yang lain.

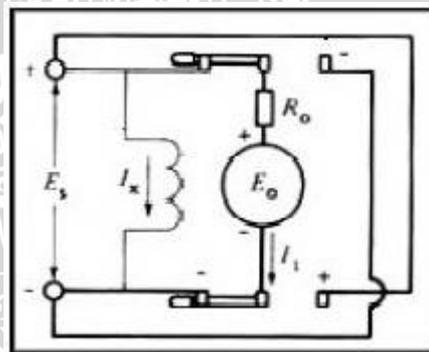


Pengereman bisa menghentikan motor bahkan lebih cepat dengan menggunakan metode yang disebut *plugging*. Ini terdiri dari tiba-tiba membalikkan arus angker dengan membalik terminal sumber.

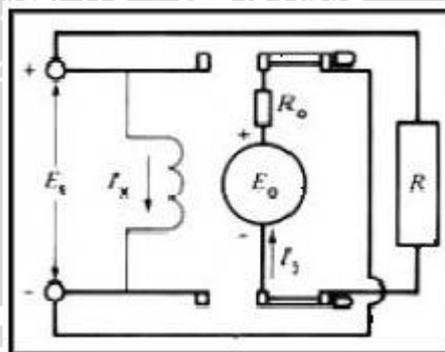
Di bawah kondisi motor normal, angker arus / 1 diberikan oleh

$$I_1 = (E_s - E_o) / R_o \dots\dots\dots(2.2)$$

di mana R_o adalah resistansi armature. Jika tiba-tiba membalik terminal sumber tegangan netto yang bekerja pada sirkuit angker menjadi $(E_o + E_s)$. Yang disebut *counter-ggl* E_o dari angker tidak lagi bertentangan dengan apa-apa tetapi sebenarnya menambah tegangan suplai E_o . Bersih ini tegangan akan menghasilkan arus balik yang sangat besar, mungkin 50 kali lebih besar daripada beban penuh arus armature.



Gambar 2.2 Amature terhubung ke sumber dc E_s .
(sumber: elektronika-dasar.web.id, 2015)



Gambar 2.3 Amature terhubung Rangkaian terbalik.
(sumber: elektronika-dasar.web.id, 2015)

Untuk mencegah suatu hal yang tidak diinginkan, harus dibatasi arus balik dengan memperkenalkan sebuah resistor R dalam seri dengan rangkaian

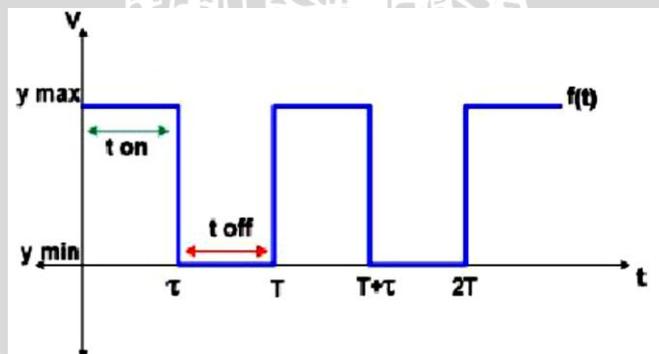


pembalikan. Seperti dalam pengereman dinamis, resistor dirancang untuk membatasi pengereman awal arus I_2 sampai sekitar dua kali arus beban penuh.

Dengan memasukkan rangkaian, torsi reverse dikembangkan bahkan ketika angker telah datang berhenti. Akibatnya, pada kecepatan nol, $E_o = 0$, tapi $I_2 = E_o / R$, yaitu sekitar satu setengah nilai awalnya. Begitu motor berhenti, harus segera dibuka sirkuit angker, selain itu akan mulai berjalan secara terbalik. Sirkuit gangguan biasanya dikontrol oleh sebuah pengontrol kecepatan otomatis perangkat yang terpasang pada poros motor.

2.2 Pulse Width Modulation (PWM)

Metode *Pulse Width Modulation* (PWM) adalah metode yang cukup efektif untuk mengendalikan kecepatan motor DC. PWM ini bekerja dengan membuat gelombang persegi yang memiliki frekuensi tetap, rasio (*duty cycle*) pulsa tinggi terhadap pulsa rendah yang telah ditentukan, yang bisa diatur dari 0% hingga 100%. Rasio pulsa tinggi terhadap pulsa rendah digunakan untuk mengatur penyambungan (*switch*) suplay tegangan terminal motor DC. Semakin besar lebar pulsa tinggi dari pada pulsa rendah berarti penyambungan suplay tegangan terminal motor mendapat porsi waktu yang lebih lama, menyebabkan kecepatan motor semakin besar.

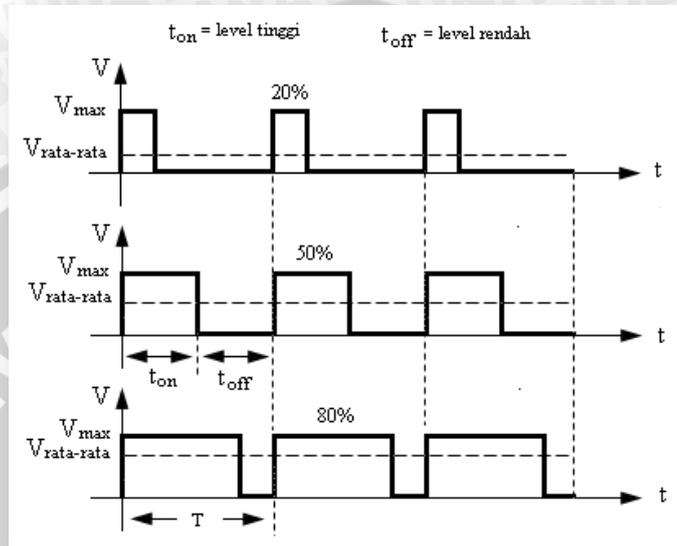


Gambar 2.4 Gelombang kotak $f(t)$ yang ideal dengan periode T

Pada Gambar 2.4 ditunjukkan gelombang kotak yang ideal dengan periode T dengan nilai T adalah jumlah dari t saat *on* dan *off*.

Modulasi lebar pulsa (*Pulse Width Modulation*) merupakan suatu teknik untuk mengendalikan tegangan input rangkaian analog dengan memanfaatkan sinyal output digital (Pulsa-pulsa tegangan digital). Pengaturan tegangan output

dapat dilakukan dengan mengubah-ubah nilai *duty cycle*, dimana *duty cycle* merupakan prosentase lebar pulsa pada kondisi aktif (level tinggi) per-periode. Besar tegangan yang dihasilkan merupakan tegangan rata-rata, hasil perkalian nilai *duty cycle* dengan tegangan maksimum sumber.



Gambar 2.5 Sinyal PWM dengan Variasi *Duty Cycle*

(Sumber: Kendali Kecepatan motor DC, TATA RISDIAN ROSDIANA, 2009)

Gambar 2.5 menunjukkan contoh pulsa PWM dengan 3 variasi nilai *duty cycle* yaitu 20%, 50% dan 80%.

Tegangan rata-rata yang diperoleh dari pulsa PWM pada Gambar 2.5 dapat dihitung dengan persamaan di bawah ini :

$$V \text{ rata-rata} = \frac{t_{on}}{T} \times V_{maks} \dots\dots\dots(2.3)$$

Karena,

$$Dc = \frac{t_{on}}{T} \times 100\% \dots\dots\dots(2.4)$$

Maka,

$$V \text{ rata-rata} = dc \times V_{maks} \dots\dots\dots(2.5)$$

Dengan :

V = tegangan

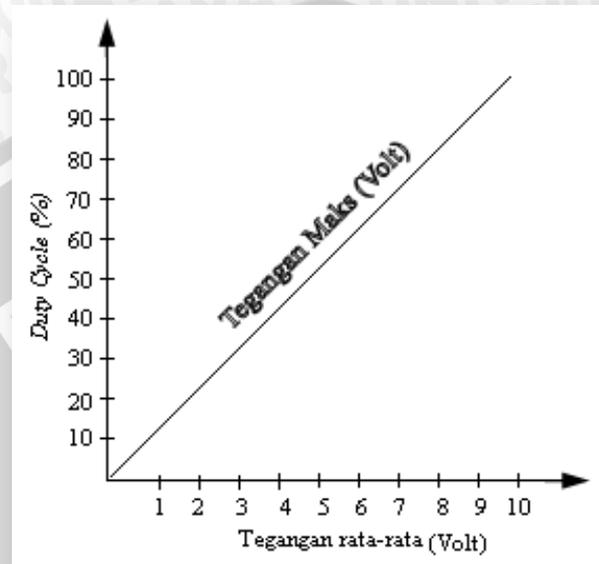
T = Periode

t on = waktu aktif (detik)



$dc = \text{duty cycle} (\%)$

Dari persamaan (2.5) diketahui bahwa tegangan maksimum linier terhadap perbandingan nilai *duty cycle* dan tegangan rata – rata, grafik perbandingan nilai *duty cycle* dan tegangan rata – rata ditunjukkan pada Gambar 2.6:



Gambar 2.6 Hubungan Nilai *Duty Cycle* dengan Tegangan Rata-rata
(Sumber: Kendali Kecepatan motor DC, TATA RISDIAN ROSDIANA, 2009)

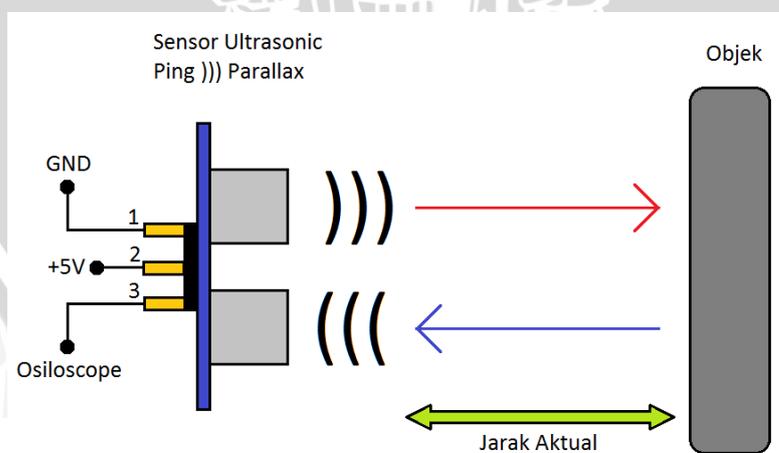
Dalam hal ini PWM diaplikasikan sebagai Pengontrolan tegangan pada Motor Listrik DC. Motor DC yang biasa ditemui di pasaran yang memiliki kutub A dan kutub B yang jika diberikan beda potensial diantara kedua-nya, maka Motor DC akan berputar. Pada prinsipnya Motor DC jenis ini akan ada waktu antara saat beda potensial diantara keduanya dihilangkan dan waktu berhentinya. Prinsip inilah yang digunakan untuk mengendalikan kecepatan Motor DC jenis ini dengan PWM, semakin besar lebar pulsa positif dari PWM maka akan semakin cepat putaran Motor DC. Untuk mendapatkan putaran Motor DC yang halus, maka perlu dilakukan penyesuaian Frekuensi (Periode Total) PWM-nya.

2.3 Sensor Ultrasonic



Gambar 2.7 Sensor PING))) /Ultrasonic
(Sumber : Parallax ,2008)

Pada Gambar 2.7 merupakan bentuk fisik dari sensor PING)))/Ultrasonic dengan dua pemancar, *transmitter* dan *receiver* yang keduanya menggunakan gelombang ultrasonik sebagai pemancarnya. Gelombang ultrasonik merupakan gelombang dengan besar frekuensi diatas frekuensi gelombang suara (*speech signals*) yaitu lebih dari 20 KHz. Seperti telah disebutkan bahwa sensor ultrasonik terdiri dari rangkaian pemancar ultrasonik yang disebut *transmitter* dan rangkaian penerima ultrasonik yang disebut *receiver*. Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari *transmitter* ultrasonik. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan dan diterima oleh *receiver* ultrasonik. Sinyal yang diterima oleh rangkaian *receiver* dikirimkan ke rangkaian mikrokontroler untuk selanjutnya diolah untuk menghitung jarak terhadap benda di depannya (bidang pantul). Untuk lebih jelas bisa dilihat pada Gambar 2.8.



Gambar 2.8 Skema sensor ultrasonik
(Sumber: PING))) parallax,mikrokontroler.org, 2015)

Rumus jarak pada sensor PING))):

$$S = \frac{V}{2} \times t \dots\dots\dots(2.6)$$

Dimana:

S = jarak yang di ukur (cm)

V= kecepatan suara 340 m/s

t = waktu yang di butuhkan (s)

Waktu yang dihitung adalah waktu pergi dan waktu datang sehingga jarak yang ditempuh adalah dua kali. Dari Gambar 2.7 sensor PING)) tersebut hanya memiliki tiga buah kaki, VCC, GROUND, dan SIGNAL. SIGNAL ini berfungsi sebagai pengatur serta penerima sensor yang akan ditangkap oleh PIR. Mengirim sinyal kasih tegangan kotak 0 1 0 menerima sinyal pantul maka ping akan jadi 0.

2.3.1 Prinsip kerja sensor ultrasonik

1. Sinyal dipancarkan oleh pemancar ultrasonik. Sinyal tersebut berfrekuensi diatas 20kHz, biasanya yang digunakan untuk mengukur jarak benda adalah 40kHz. Sinyal tersebut di bangkitkan oleh rangkaian pemancar ultrasonik.
2. Sinyal yang dipancarkan tersebut kemudian akan merambat sebagai sinyal / gelombang bunyi dengan kecepatan bunyi yang berkisar 340 m/s. Sinyal tersebut kemudian akan dipantulkan dan akan diterima kembali oleh bagian penerima Ultrasonik.
3. Setelah sinyal tersebut sampai di penerima ultrasonik, kemudian sinyal tersebut akan diproses untuk menghitung jaraknya.

Jarak dihitung berdasarkan rumus :

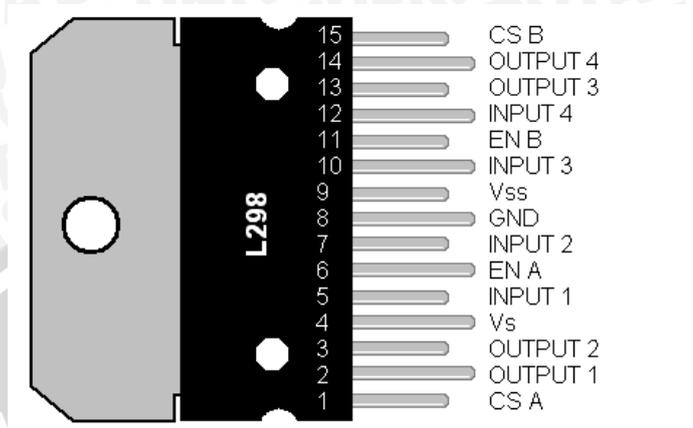
$$S = 340.t/2$$

dimana S adalah jarak antara sensor ultrasonik dengan bidang pantul, dan t adalah selisih waktu antara pemancaran gelombang ultrasonik sampai diterima kembali oleh bagian penerima ultrasonik.

2.4 Driver Motor

Untuk *Driver Motor* digunakan IC L298, Rangkaian *driver* motor DC dengan IC L298 dapat digunakan untuk mengendalikan motor DC dengan arus maksimum hingga 4A. Dengan IC *driver* motor DC L298 dapat digunakan untuk

mngendalikan 2 buah motor DC sekaligus secara independent. Kemampuan tiap *driver* motor DC dalam IC L298 ini adalah 4A untuk masing-masing *driver*. IC L298 adalah *driver* motor DC H-Bridge dengan 2 unit *driver* didalam 1 chip IC.



Gambar 2.6 Konfigurasi Pin IC L298

(sumber: STMicroelectronics, 2000)

Keterangan Gambar 2.6 sebagai berikut:

Pada Input 1, input 2, input 3, input 4 dihubungkan ke mikrokontroler sebagai input direksi Enable A, Enable B dihubungkan ke port OC pada mikro sebagai input PWM Cur Sen A, Cur Sen B dihubungkan ke ground Logic Supply Voltage Vss dihubungkan ke vcc mikro (5V) Supply Voltage Vs dihubungkan ke Catu Daya batere (12V) Output 1, output 2, output 3, output 3 dihubungkan ke motor DC.

Feature yang dimiliki IC *driver* motor DC L298 sesuai datasheet adalah :

- Mampu menampung *Voltage Supply* hingga 46V
- Total *Dc Current* hingga 4A.
- *Low Saturation Voltage*.
- *Overtemperature Protection*.
- *Logical "0" Input Voltage* sampai 1,5V
- *High Noise Immunity*.

2.5 Arduino Uno

Uno Arduino adalah *board* berbasis mikrokontroler pada ATmega328. *Board* ini memiliki 14 digital *input / output* pin (di mana 6 pin dapat digunakan sebagai *output* PWM), 6 *input* analog, 16 MHz osilator kristal, koneksi USB, jack

listrik tombol reset. Pin-pin ini berisi data yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya.

Board Arduino Uno memiliki fitur-fitur baru sebagai berikut :

- 1,0 pin *out*: tambah SDA dan SCL pin yang dekat ke pin aref dan dua pin baru lainnya ditempatkan dekat ke pin RESET, dengan IO REF yang memungkinkan sebagai *buffer* untuk beradaptasi dengan tegangan yang disediakan dari *board* sistem. Pengembangannya, sistem akan lebih kompatibel dengan Prosesor yang menggunakan AVR, yang beroperasi dengan 5V karena Arduino yang beroperasi dengan 3.3V. Yang kedua adalah pin tidak terhubung, yang disediakan untuk tujuan pengembangannya. Bentuk keseluruhan *Board* Arduino UNO ditunjukkan pada Gambar 2.9 dan kabel USB penghubung untuk *Board* Arduino UNO ditunjukkan pada Gambar 2.10.



Gambar 2.9 Board Arduino UNO
(sumber: arduino.cc, 2015)



Gambar 2.10 Kabel Usb Board Arduino UNO

(sumber: arduino.cc, 2015)

Datasheet dari Arduino UNO ditampilkan pada Tabel 2.1.

Tabel 2.1 Deskripsi Arduino Uno

(sumber: arduino.cc, 2015)

<i>Mikrokontroler</i>	Atmega328
<i>Operasi Voltage</i>	5V
<i>Input Voltage</i>	7-12 V (Rekomendasi)
<i>Input Voltage</i>	6-20 V (limits)
<i>I/O</i>	14 pin (6 pin untuk PWM)
<i>Arus</i>	50 mA
<i>Flash Memory</i>	32KB
<i>Bootloader</i>	SRAM 2 KB
<i>EEPROM</i>	1 KB
<i>Kecepatan</i>	16 Mhz

2.5.1 Catu Daya

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

Board dapat beroperasi pada pasokan daya dari 6 - 20 volt. Jika diberikan dengan kurang dari 7V, bagaimanapun, pin 5V dapat menyuplai kurang dari 5 volt dan *board* mungkin tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak *board*. Rentang yang dianjurkan adalah 7 - 12 volt.

Pin catu daya adalah sebagai berikut:

- VIN. Tegangan *input* ke *board* Arduino ketika menggunakan sumber daya eksternal (sebagai lawan dari 5 volt dari koneksi USB atau sumber daya lainnya diatur). Anda dapat menyediakan tegangan melalui pin ini, atau, jika memasok tegangan melalui colokan listrik, mengaksesnya melalui pin ini.
- 5V. Catu daya diatur digunakan untuk daya mikrokontroler dan komponen lainnya di *board*. Hal ini dapat terjadi baik dari VIN melalui regulator *onboard*, atau diberikan oleh USB .

- 3,3 volt pasokan yang dihasilkan oleh regulator *on-board*. Menarik arus maksimum adalah 50 mA.
- GND

2.5.2 Memory ATmega328

ATMega328 merupakan mikrokontroler keluarga AVR 8 bit. Beberapa tipe mikrokontroler yang sama dengan ATMega8 ini antara lain ATMega8535, ATMega16, ATMega32, ATmega328, yang membedakan antara mikrokontroler antara lain adalah, ukuran memori, banyaknya GPIO (pin *input/output*), *peripheral* (USART, timer, counter, dll). Dari segi ukuran fisik, ATMega328 memiliki ukuran fisik lebih kecil dibandingkan dengan beberapa mikrokontroler diatas. Namun untuk segi memori dan *peripheral* lainnya ATMega328 tidak kalah dengan yang lainnya karena ukuran memori dan periperialnya relatif sama dengan ATMega8535, ATMega32, hanya saja jumlah GPIO lebih sedikit dibandingkan mikrokontroler Atmega328.

ATmega 328

(PCINT14/RESET) PC6	1	[A5]28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2[0] ^{RX}	[A4]27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3[1] ^{TX}	[A3]26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4[2]	[A2]25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5[3]~	[A1]24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6[4]	[A0]23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	[13]19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11[5]~	[12]18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12[6]~	[11]17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13[7]	[10]16	PB2 (\overline{SS} /OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14[8]	[9]15	PB1 (OC1A/PCINT1)

~ = PWM

Gambar 2.11 Konfigurasi Pin Atmega 328

(sumber: Atmel, Atmega328, 2014)

Pada Gambar 2.11 ATMega328 memiliki 3 buah PORT utama yaitu PORTB, PORTC, dan PORTD dengan total pin *input/output* sebanyak 23 pin. PORT tersebut dapat difungsikan sebagai *input/output* digital atau difungsikan sebagai periperial lainnya.

Bagian tiap-tiap port:



1. Port B

Port B merupakan jalur data 8 bit yang dapat difungsikan sebagai input/output. Selain itu PORTB juga dapat memiliki fungsi alternatif seperti di bawah ini.

- a. ICP1 (PB0), berfungsi sebagai *Timer Counter 1 input capture* pin.
- b. OC1A (PB1), OC1B (PB2) dan OC2 (PB3) dapat difungsikan sebagai keluaran PWM (*Pulse Width Modulation*).
- c. MOSI (PB3), MISO (PB4), SCK (PB5), SS (PB2) merupakan jalur komunikasi SPI.
- d. Selain itu pin ini juga berfungsi sebagai jalur pemrograman serial (ISP).
- e. TOSC1 (PB6) dan TOSC2 (PB7) dapat difungsikan sebagai sumber *clock external* untuk *timer*.
- f. XTAL1 (PB6) dan XTAL2 (PB7) merupakan sumber *clock* utama mikrokontroler.

2. Port C

Port C merupakan jalur data 7 bit yang dapat difungsikan sebagai *input/output* digital. Fungsi alternatif PORTC antara lain sebagai berikut:

- a. ADC6 *channel* (PC0,PC1,PC2,PC3,PC4,PC5) dengan resolusi sebesar 10 bit. ADC dapat digunakan untuk mengubah *input* yang berupa tegangan analog menjadi data digital.
- b. I2C (SDA dan SDL) merupakan salah satu fitur yang terdapat pada PORTC. I2C digunakan untuk komunikasi dengan sensor atau *device* lain yang memiliki komunikasi data tipe I2C seperti sensor kompas, *accelerometer nunchuck*.

3. Port D

Port D merupakan jalur data 8 bit yang masing-masing pin-nya juga dapat difungsikan sebagai *input/output*. Sama seperti Port B dan Port C, Port D juga memiliki fungsi alternatif di bawah ini:

- a. USART (TXD dan RXD) merupakan jalur data komunikasi serial dengan level sinyal TTL. Pin TXD berfungsi untuk mengirimkan data serial, sedangkan RXD kebalikannya yaitu sebagai pin yang berfungsi untuk menerima data serial.
- b. *Interrupt* (INT0 dan INT1) merupakan pin dengan fungsi khusus sebagai interupsi *hardware*. Interupsi biasanya digunakan sebagai selaan dari program, misalkan pada saat program berjalan kemudian terjadi interupsi *hardware/software* maka program utama akan berhenti dan akan menjalankan program interupsi.
- c. XCK dapat difungsikan sebagai sumber *clock external* untuk USART, namun juga dapat memanfaatkan *clock* dari CPU, sehingga tidak perlu membutuhkan *external clock*.
- d. T0 dan T1 berfungsi sebagai masukan *counter external* untuk *timer 1 dan timer 0*.
- e. AIN0 dan AIN1 keduanya merupakan masukan *input* untuk *analog comparator*.

2.5.3 Input & Output Arduino UNO

Masing-masing dari 14 pin digital pada Uno dapat digunakan sebagai *input* atau *output*, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Mereka beroperasi di 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA.

Beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data TTL serial. Pin ini terhubung ke pin yang sesuai dari chip ATmega8U2 USB-to-Serial TTL.
- Eksternal Interupsi: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, tepi naik atau jatuh, atau perubahan nilai. Lihat `attachInterrupt ()` fungsi untuk rincian.
- PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan 8-bit *output* PWM dengan `analogWrite ()` fungsi.

- SPI: 10 (SS), 11 (mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI.
- LED: 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin adalah nilai *high*, LED menyala, ketika pin adalah nilai *low*, itu *off*.

Arduino Uno memiliki 6 *input analog*, diberi label A0 melalui A5, masing-masing menyediakan 10 bit resolusi yaitu 1024 nilai yang berbeda. Secara *default* sistem mengukur dari tanah sampai 5 volt.

- TWI: A4 atau SDA pin dan A5 atau SCL pin. Mendukung komunikasi TWI
- Aref. Referensi tegangan untuk *input analog*. Digunakan dengan *analog Reference()*.
- Reset.

2.5.4 Komunikasi Arduino UNO

Uno Arduino memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). Sebuah ATmega16U2 pada saluran board ini komunikasi serial melalui USB dan muncul sebagai com port virtual untuk perangkat lunak pada komputer. *Firmware* Arduino menggunakan USB *driver* standar COM, dan tidak ada *driver* eksternal yang dibutuhkan. Namun, pada Windows, file. *Inf* diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke *board* Arduino UNO. RX dan TX LED di *board* akan berkedip ketika data sedang dikirim melalui *chip* USB-to-serial dan koneksi USB ke komputer. ATmega328 ini juga mendukung komunikasi I2C (TWI) dan SPI. Fungsi ini digunakan untuk melakukan komunikasi *inteface* pada sistem.

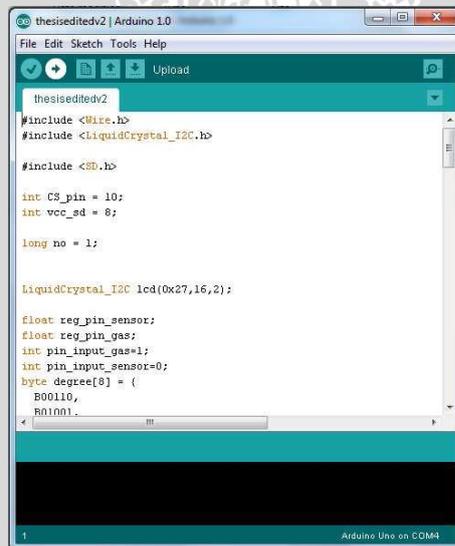
2.5.5 Programming

Arduino UNO dapat diprogram dengan perangkat lunak Arduino. Pilih Arduino Uno dari *Tool* lalu sesuaikan dengan mikrokontroler yang digunakan. Para ATmega328 pada Uno Arduino memiliki *bootloader* yang memungkinkan Anda untuk meng-*upload* program baru untuk itu tanpa menggunakan

programmer hardware eksternal. Untuk berkomunikasi menggunakan protokol dari bahasa C. Sistem dapat menggunakan perangkat lunak FLIP Atmel (Windows) atau *programmer DFU* (Mac OS X dan Linux) untuk memuat *firmware* baru. Atau Anda dapat menggunakan *header ISP* dengan eksternal *programmer*.

2.5.6 Perangkat Lunak (Arduino IDE)

Lingkungan *open-source* Arduino memudahkan untuk menulis kode dan meng-*upload* ke *board* Arduino. Ini berjalan pada Windows, Mac OS X, dan Linux. Berdasarkan Pengolahan, *avr-gcc*, dan perangkat lunak sumber terbuka lainnya. Contoh gambar pada tampilan perangkat lunak Arduino IDE bisa di lihat pada Gambar 2.9.



Gambar 2.9 Tampilan Framework Arduino UNO

(sumber: software arduino 1.6.5, 2015)

2.5.7 Otomatis Software Reset

Tombol reset Uno Arduino dirancang untuk menjalankan program yang tersimpan didalam mikrokontroler dari awal. Tombol reset terhubung ke Atmega328 melalui kapasitor 100nf. Setelah tombol reset ditekan cukup lama untuk me-reset *chip*, *software* IDE Arduino dapat juga berfungsi untuk meng-*upload* program dengan hanya menekan tombol upload di *software* IDE Arduino.

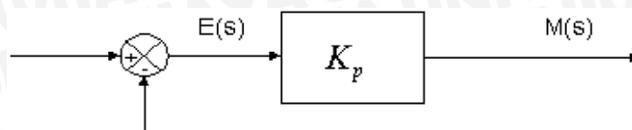
2.6 Kontroler Proporsional Integral Derivative (PID)

Di dalam suatu sistem kontrol dikenal adanya beberapa macam aksi kontrol, diantaranya yaitu aksi kontrol proporsional, aksi kontrol integral dan aksi kontrol derivative. Masing-masing aksi kontrol ini mempunyai keunggulan-keunggulan tertentu, dimana aksi kontrol proporsional mempunyai keunggulan *rise time* yang cepat, aksi kontrol integral mempunyai keunggulan untuk memperkecil *error*, dan aksi kontrol *derivative* mempunyai keunggulan untuk memperkecil *error* atau meredam *overshoot/undershoot*. Untuk itu agar dapat menghasilkan *output* dengan *risetime* yang cepat dan *error* yang kecil dapat menggabungkan ketiga aksi kontrol ini menjadi aksi kontrol PID. Parameter pengontrol *Proporsional Integral derivative* (PID) selalu didasari atas tinjauan terhadap karakteristik yang di atur (*plant*). Dengan demikian bagaimanapun rumitnya suatu *plant*, perilaku *plant* tersebut harus di ketahui terlebih dahulu sebelum pencarian parameter PID itu dilakukan.

2.6.1 Pengontrol proporsional

Pengontrol proporsional memiliki keluaran yang sebanding atau proporsional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Secara lebih sederhana dapat dikatakan bahwa keluaran pengontrol proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan pada sinyal masukan akan segera menyebabkan sistem secara langsung mengeluarkan *output* sinyal sebesar konstanta pengalinya.

Gambar 2.10 menunjukkan blok diagram yang menggambarkan hubungan antara besaran setting, besaran aktual dengan besaran keluaran pengontrol proporsional. Sinyal kesalahan (*error*) merupakan selisih antara besaran setting dengan besaran aktualnya. Selisih ini akan mempengaruhi pengontrol, untuk mengeluarkan sinyal positif (mempercepat pencapaian harga setting) atau negatif (memperlambat tercapainya harga yang diinginkan).



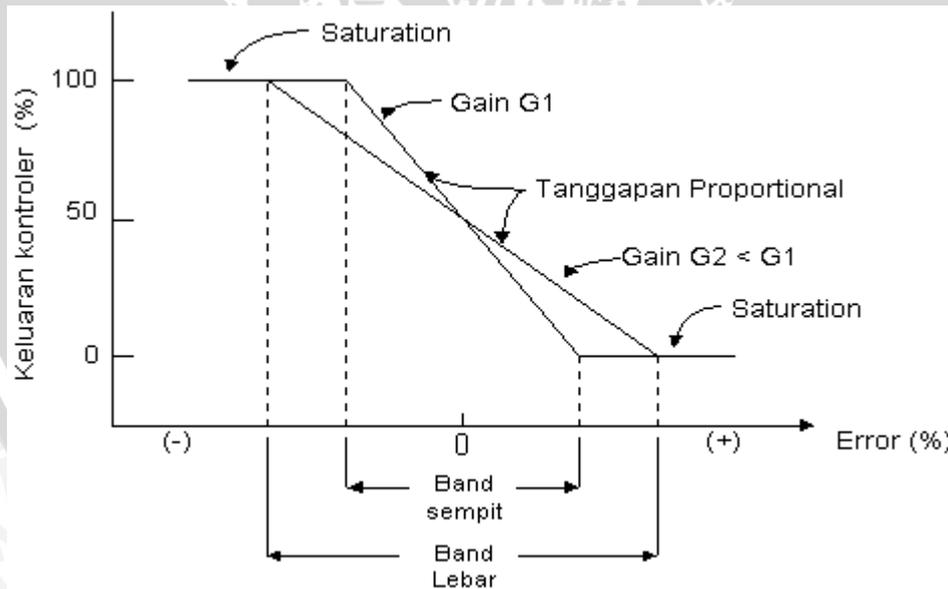
Gambar 2.10 Diagram blok kontroler proporsional

(Sumber: Ogata K., 1997)

pengontrol proporsional memiliki 2 parameter, pita proporsional (*propotional band*) dan konstanta proporsional. Daerah kerja kontroler efektif dicerminkan oleh pita proporsional sedangkan konstanta proporsional menunjukkan nilai faktor penguatan sinyal terhadap sinyal kesalahan K_p . Hubungan antara pita proporsional (PB) dengan konstanta proporsional (K_p) ditunjukkan secara persentasi oleh persamaan berikut:

$$PB = \frac{1}{K_p} \times 100\% \dots\dots\dots(2-7)$$

Pada Gambar 2.11 menunjukkan grafik hubungan antara PB, keluaran pengontrol dan kesalahan yang merupakan masukan pengontrol. Ketika konstanta proporsional bertambah semakin tinggi, pita proporsional menunjukkan penurunan yang semakin kecil, sehingga lingkup kerja yang dikuatkan akan semakin sempit.



Gambar 2.11 Proportional band dari pengontrol proporsional tergantung pada penguatan.

(sumber : elektronikaIndonesia, 1998)



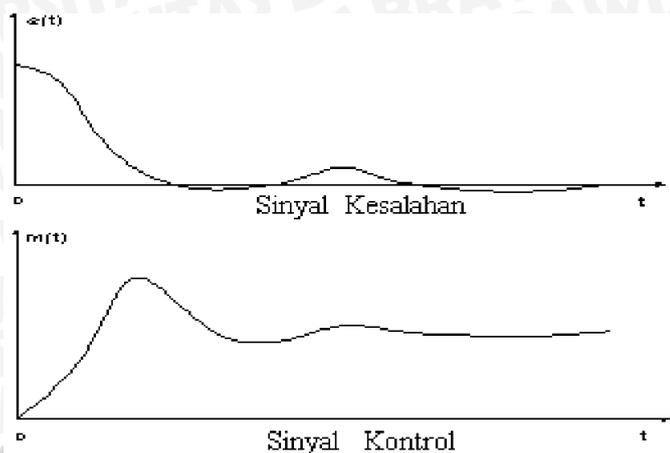
Ciri-ciri pengontrol proposional harus diperhatikan ketika pengontrol tersebut diterapkan pada suatu sistem. Secara eksperimen, pengguna pengontrol proposional harus memperhatikan ketentuan-ketentuan berikut ini :

1. kalau nilai K_p kecil, pengontrol proposional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat.
2. kalau nilai K_p dinaikan, respon sistem menunjukkan semakin cepat mencapai set point dan keadaan stabil.
3. namun jika nilai K_p diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan sistem bekerja tidak stabil, atau respon sistem akan berosolasi.

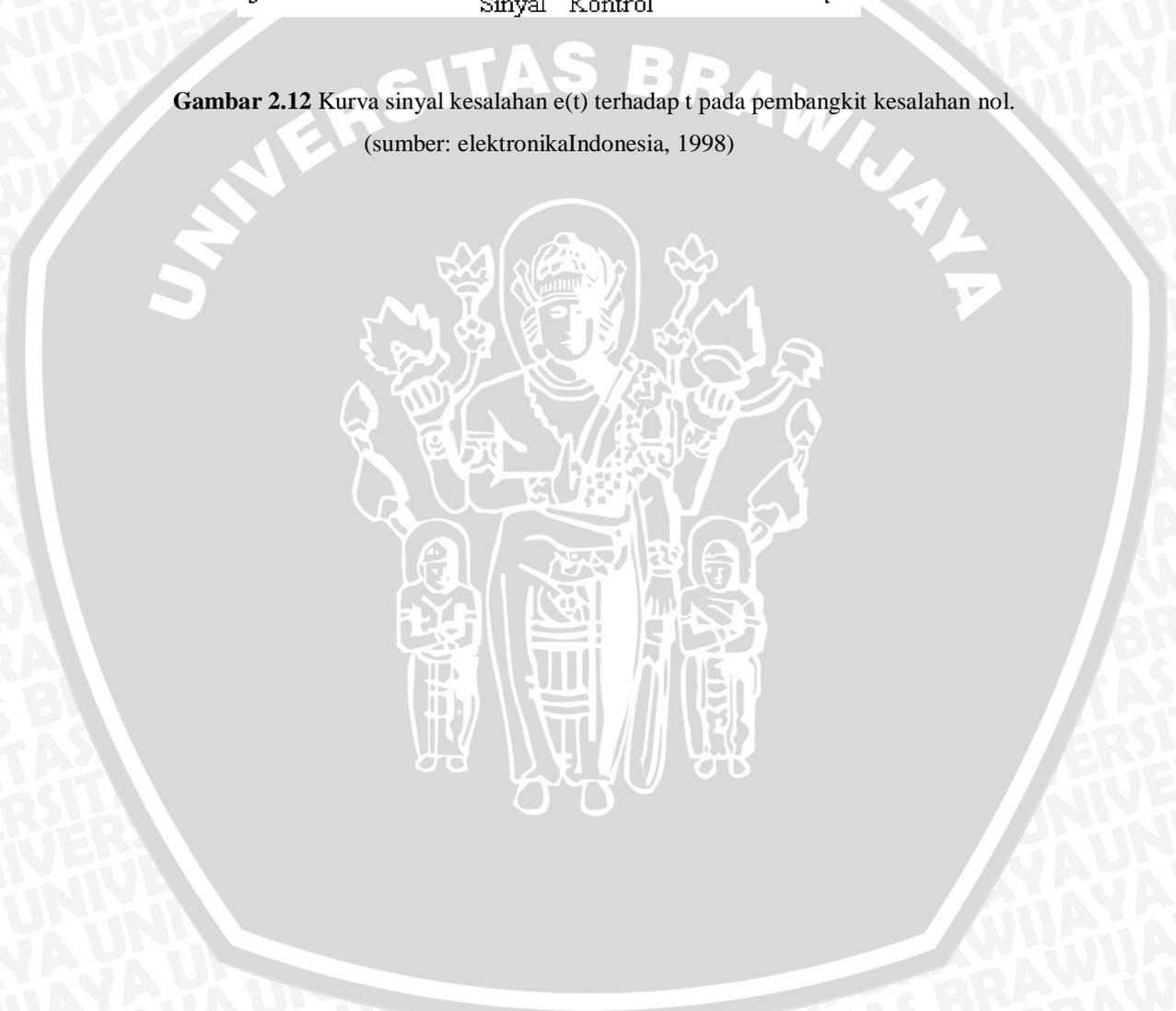
2.6.2 Pengontrol Integral

Pengontrol integral berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan stabil nol. Jika sebuah plant tidak memiliki unsur integrator ($1/s$), pengontrol proposional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan stabilnya nol. Dengan pengontrol integral, respon sistem dapat diperbaiki, yaitu mempunyai kesalahan keadaan stabilnya nol. Pengontrol integral memiliki karakteristik seperti halnya sebuah integral. Keluaran sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran pengontrol ini merupakan penjumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Sinyal keluaran pengontrol integral merupakan luas bidang yang dibentuk oleh kurva kesalahan penggerak. Sinyal keluaran akan berharga sama dengan harga sebelumnya ketika sinyal kesalahan berharga nol.

Gambar 2.12 menunjukkan contoh sinyal kesalahan yang dimasukkan ke dalam pengontrol integral dan keluaran pengontrol integral terhadap perubahan sinyal kesalahan tersebut.



Gambar 2.12 Kurva sinyal kesalahan $e(t)$ terhadap t pada pembangkit kesalahan nol.
 (sumber: elektronikaIndonesia, 1998)

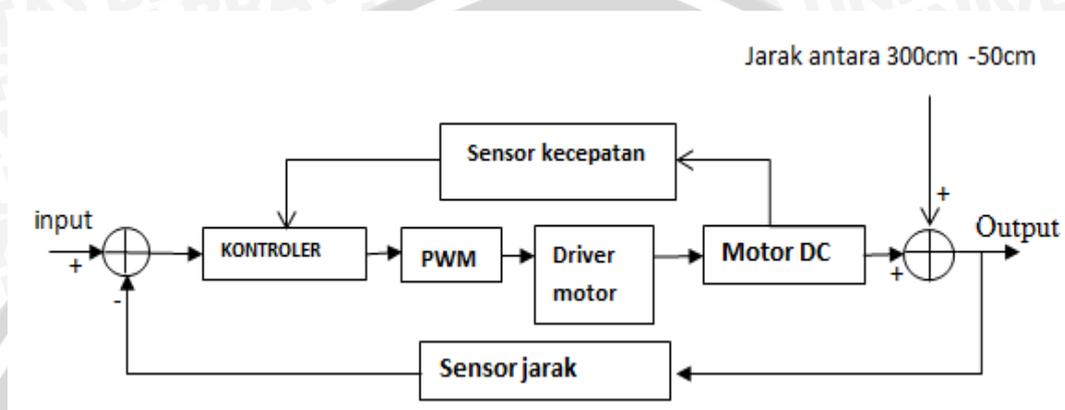


BAB III

METODE PENELITIAN

Perancangan ini didasarkan pada masalah yang bersifat aplikatif, yaitu perencanaan dan perealisasiian alat agar dapat bekerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah.

Untuk diagram blok sistem secara keseluruhan dapat dilihat pada Gambar 3.1.

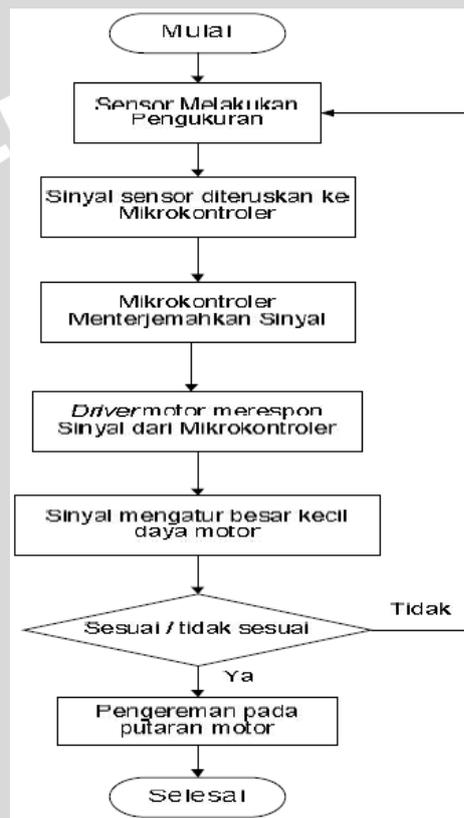


Gambar 3.1 Diagram Blok sistem
(sumber: perancangan 2015)

Perancangan dimulai dengan menghidupkan motor DC dengan tegangan masukannya sebesar 12 VDC. Dimana 12Volt adalah tegangan yang telah ditentukan pada alat sebagai tegangan maksimum dari prototipe ini. Motor DC sudah terhubung dengan beberapa komponen yang berfungsi sebagai rangkaian pengendali putaran, Dengan menggunakan sensor ultrasonik yang berfungsi sebagai indikator jarak benda yang mendekat yang kemudian akan memberikan informasi ke mikrokontroler dengan pin *input/output*, masukan dari sensor ultrasonik akan diproses secara digital yang kemudian akan dikonfigurasi dengan pin 6 yang berfungsi sebagai pembangkit sinyal *Pulse Width Modulation* (PWM). Sinyal PWM berupa *Dutty cycle* dari tegangan *output*. Pin 6 akan terhubung dengan rangkaian *driver* motor, dengan mengatur lebar pulsa PWM, maka motor akan dapat diatur kecepatannya. Lebar pulsa yang dikeluarkan oleh mikrokontroler berupa sinyal *high* dan *low* tegangan, sehingga yang diatur adalah besar kecilnya tegangan yang dimasukkan ke kaki IC L298. Sehingga masukan dari sumber akan berbanding lurus dengan besar pulsa dari kontroler.

3.1 Diagram Alir Prinsip kerja Alat

Berikut adalah diagram alir Penelitian, dimana proses dimulai dari mikrokontroler yang men-*trigger* sensor ultrasonik ping sehingga mampu melakukan pengukuran terhadap benda pantul yang berada di depannya, sinyal kemudian diterima kembali oleh mikrokontroler untuk kemudian di eksekusi. Hasil pengukuran kemudian diterjemahkan mikrokontroler untuk sebagai masukan *driver* motor. Masukan dari mikrokontroler akan mengatur besar kecilnya sumber tegangan yang diterima motor DC dengan menggunakan metode pwm. Pada Gambar 3.2 ditunjukkan diagram alir pada prinsip kerja alat.



Gambar 3.2 Diagram Alir Sistem Kerja Alat
(sumber: Perancangan 2015)

3.2 Perancangan dan Pembuatan Alat

Perancangan dan pembuatan alat dalam penelitian ini dibagi menjadi 2 bagian yaitu elektrik, dan *software*.

3.3 Perancangan Elektrik

Secara garis besar perancangan perangkat keras (*Hardware*) dibagi dalam beberapa tahap sebagai berikut:

- 1) Penentuan spesifikasi alat.
- 2) Pembuatan blok diagram keseluruhan sistem.
- 3) Penentuan komponen yang akan digunakan.
- 4) Merakit perangkat keras masing-masing blok.

3.4 Perancangan dan Pembuatan Perangkat Lunak (*Software*)

Penyusunan perangkat lunak (*software*) digunakan untuk mengetahui kondisi – kondisi yang ingin di desain dan parameter yang telah dirancang kemudian diterapkan ke dalam mikrokontroler.

3.5 Pengujian dan Analisis

Untuk menganalisis kinerja alat apakah sesuai dengan yang direncanakan maka dilakukan pengujian sistem. Pengujian dilakukan pada mikrokontroler serta Pengujian Kuat Sinyal dari Sensor.

3.6 Pengambilan Kesimpulan dan Saran

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 penyempurnaannya.

3.7 SISTEMATIKA PEMBAHASAN

Skripsi ini terdiri dari enam bab dengan sistematika pembahasan sebagai berikut:

BAB I Pendahuluan

Membahas latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika pembahasan.

BAB II Tinjauan Pustaka

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan alat.

BAB III Metodologi Penulisan

Membahas metode penelitian dan perencanaan alat.

BAB IV Perancangan dan Pembuatan Alat

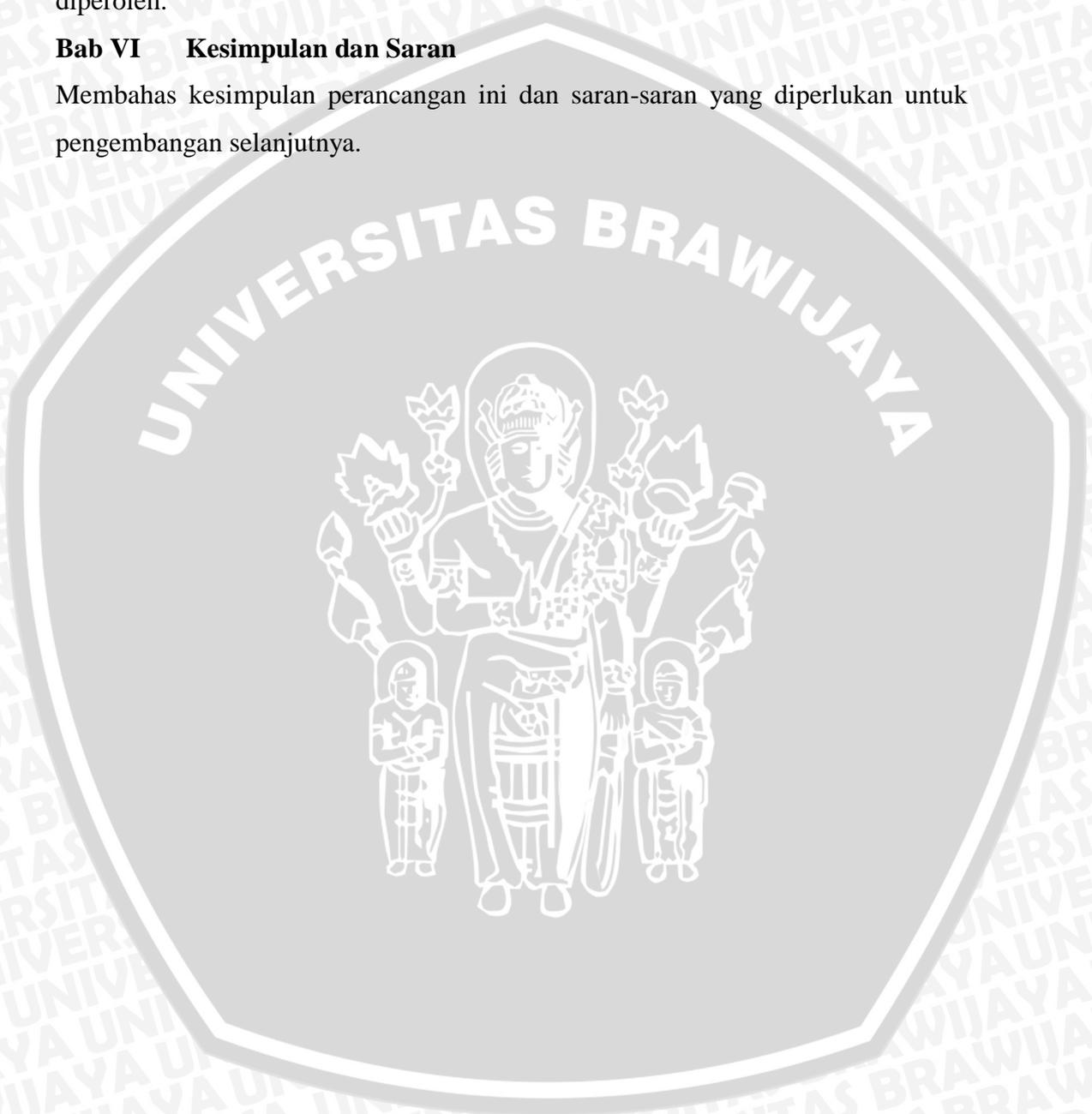
Membahas perancangan mikrokontroler Atmega16 dan menerapkannya ke dalam *software* di komputer, sehingga sistem dapat bekerja dengan baik.

BAB V Pengujian dan Analisis

Membahas hasil pengujian sistem yang sudah dibuat, serta analisis hasil yang diperoleh.

Bab VI Kesimpulan dan Saran

Membahas kesimpulan perancangan ini dan saran-saran yang diperlukan untuk pengembangan selanjutnya.



BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

Bab ini menjelaskan mengenai spesifikasi alat, perancangan perangkat keras dari sistem pengatur kecepatan motor untuk pengereman adanya penghalang yang meliputi diagram blok sistem, cara kerja sistem, dan gambar rancangan alat. Selain perancangan perangkat keras akan dijelaskan juga mengenai perancangan perangkat lunak meliputi perncangan flowchart atau diagram alir kerja system dan PID.

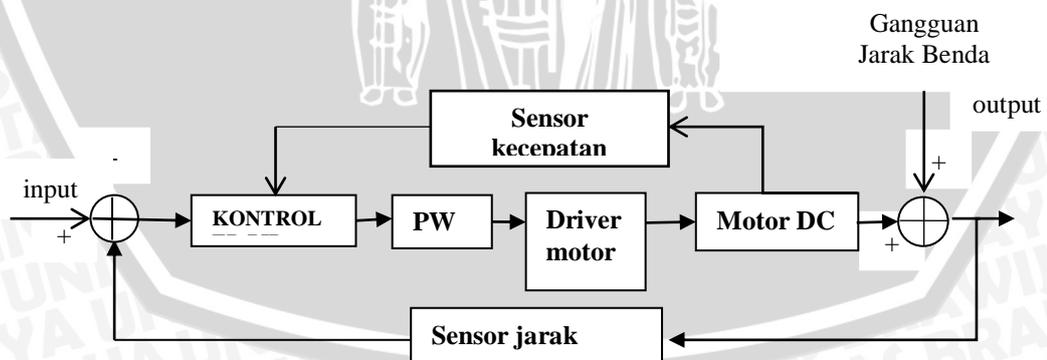
Perancangan ini terdiri dari :

- Diagram Blok Sistem
- Desain Mekanik alat
- Perancangan perangkat keras
- Perancangan perangkat lunak

Perancangan dan pembuatan dilakukan secara bertahap dan sistematis, yang nantinya akan memudahkan dalam analisa sistem.

4.1 BLOK DIAGRAM SISTEM

Pada perancangan alat diperlukan perancangan blok diagram sistem yang diharapkan agar alat dapat bekerja sesuai dengan apa yang di inginkan. Pada Gambar 4.1 akan ditunjukkan blok diagram sistem yang digunakan.



Gambar 4.1. Blok Diagram Sistem

(Sumber : Perancangan 2015)

Komponen dari sistem perancangan blok diagram pada gambar 4.1 akan dijelaskan sebagai berikut:

- Input (set point) :Kecepatan pada posisi tertentu
- Kontroler :Kontroler yang di gunakan adalah PID menggunakan Arduio dan nilai PID digunakan sebagai referensi pemanggilan nilai kontrol kecepatan motor DC untuk melakukan pengereman di depan penghalang.
- Aktuator : Motor DC 12V dengan 1300Rpm
- Sensor : Sensor PING sebagai deteksi adanya penghalang. Sinyal Feddback dari sensor rotary encoder berupa pembacaan Putaran motor DC.
- Plant : Perlambatan motor pada Posisi tertentu.
- *Driver* Motor : Menggunakan IC L928N sebagai pengendali motor DC
- Output : Perlambatan Motor Terhadap respon yang sudah dtentukan.

4.2 Prinsip Kerja Alat

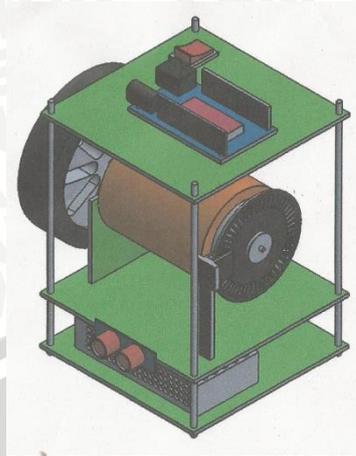
Miniatur sistem pengereman motor DC akan aktif saat tombol ON diaktifkan, kecepatan motor di setting pada putaran *steady* tertentu, dan sistem akan membaca jarak penghalang yang ada di depan melalui sensor PING))), apabila jarak kurang dari 300cm, kontroler akan mengurangi kecepatan pada motor DC sampai berhenti apabila jarak kurang atau sama dengan 50cm. Apabila jarak penghalang lebih dari 300cm motor akan tetap berputar tanpa mengalami perlambatan.

4.3 Perancangan Perangkat Keras (*Hardware*)

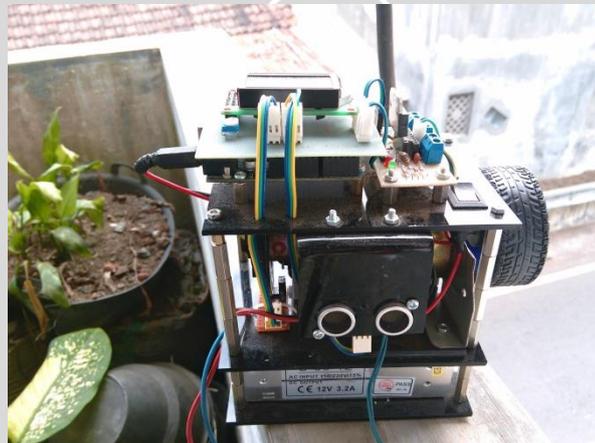
4.3.1 Perancangan Mekanik Sistem

Perancangan mekanik sistem adalah sebagai berikut:

Gambar 4.2 adalah desain mekanik yang di gambar dengan menggunakan *software 3D CAD Solidworks* dan Gambar 4.3 adalah gambar hasil pembuatan alat.



Gambar 4.2 Desain 3D Mekanik alat keseluruhan
(sumber: Perancangan, 2015)



Gambar 4.3 Gambar keseluruhan alat
(Sumber : perancangan, 2015)

Spesifikasi sistem yang dirancang adalah sebagai berikut:

1. Alat *prototype* sistem pengereman menggunakan rangka dengan spesifikasi sebagai berikut:
 - Dimensi alat secara keseluruhan yaitu panjang 13cm, lebar 10cm dan tinggi 13 cm.
 - Menggunakan 3 buah *Acrylic* dengan ukuran panjang 13cm dan lebar 10cm.

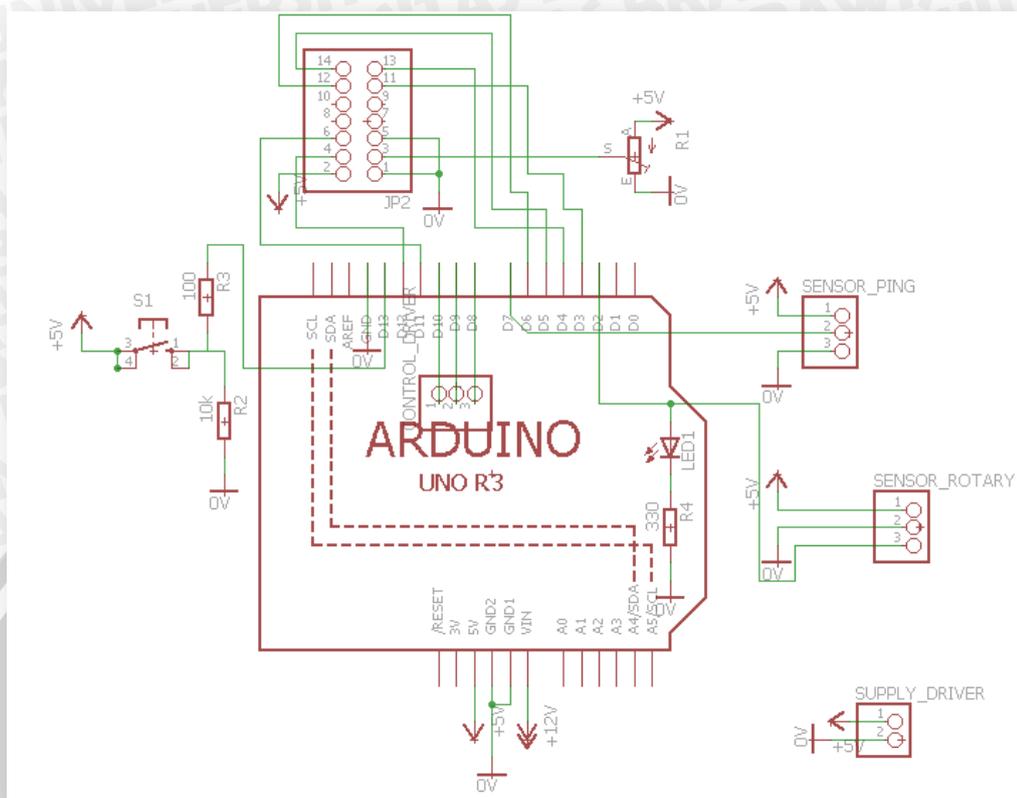
2. Pergerakan alat Berbeban menggunakan motor DC dengan 1800rpm dan catu daya 12 volt.
3. *Rotary Encoder* sebagai sensor kecepatan yang dipasang pada Motor DC.
4. Sebuah sensor PING))) digunakan sebagai sensor pendeteksi adanya penghalang.
5. Kontroler yang digunakan adalah Arduino UNO Dengan menggunakan *driver* L928N sebagai *driver* motor.
6. Catu tegangan menggunakan *Power Supply* 12 volt 3,2 A.

4.4 Perancangan Sistem Kerja

Perancangan dimulai dengan menghidupkan motor DC dengan tegangan masukannya sebesar 12 VDC. Dimana 12Volt adalah tegangan yang telah ditentukan pada alat sebagai tegangan maksimum dari prototipe ini. Motor DC sudah terhubung dengan beberapa komponen yang berfungsi sebagai rangkaian pengendali putaran, Dengan menggunakan sensor ultrasonik yang berfungsi sebagai indikator jarak benda yang mendekat yang kemudian akan memberikan informasi ke kontroler dengan pin *input/output*, masukan dari sensor ultrasonik akan diproses secara digital yang kemudian akan dikonfigurasi dengan pin D7 yang berfungsi sebagai pembangkit sinyal *Pulse Width Modulation* (pwm). Sinyal pwm berupa *Duty cycle* dari tegangan *output*. Pin tersebut akan terhubung dengan rangkaian *driver* motor, dengan mengatur lebar pulsa PWM, maka motor akan dapat diatur kecepatannya. Lebar pulsa yang dikeluarkan oleh kontroler berupa sinyal *high* dan *low* tegangan, sehingga yang diatur adalah besar kecilnya tegangan yang dimasukan ke kaki IC L298N. Sehingga masukan dari sumber akan berbanding lurus dengan besar pulsa dari mikrokontroler.

4.5 Perancangan Catu Daya Sistem

Sistem ini Menggunakan satu jenis catu daya. Catu daya 12V untuk kontroler Arduino UNO, dan juga 12V untuk *supply driver* Motor yang di kendalikan oleh IC L298N keduanya yang bersumber dari *power supply* 12V 3,2A. Skema rangkaian catu daya ditunjukkan dalam Gambar 4.4



Gambar 4.4 Skema catu daya sistem
(Sumber: perancangan, 2015)

4.6 Perancangan Sensor Ultrasonik PING))

Sensor ultrasonik pada perancangan ini berfungsi sebagai sensor jarak antara robot dengan penghalang. Keluaran dari sensor ultrasonik ini digunakan sebagai masukan (set point) dari kontroler PID. Dalam perancangan ini, digunakan satu buah sensor ultrasonik yang diletakkan di bagian depan robot dan dihubungkan dengan kontroler Arduino UNO pada pin D7.

4.7 Perancangan Perangkat Lunak

Pada penelitian ini, perancangan perangkat lunak menggunakan program Arduino 1.0.6 dengan pencarian PID menggunakan metode *trial and error*. Yaitu dengan cara melihat respon motor DC saat Arduino membaca jarak yg di dapat dari sensor PING)) dan motor bergerak berdasarkan jarak ditetapkan sebelumnya. Setelah dilakukan beberapa kali pengujian maka didapatkan parameter PID yang

paling baik yaitu $K_p = 0.02$, $K_i = 0$ dan $K_d = 0,054$. Kerangka perangkat lunak yang dibuat sesuai dengan *flowchart* dalam Gambar 4.6.



Gambar 4.6 Diagram alir sistem Perangkat lunak

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 awal. Pengujian dibagi menjadi beberapa bagian, yaitu:

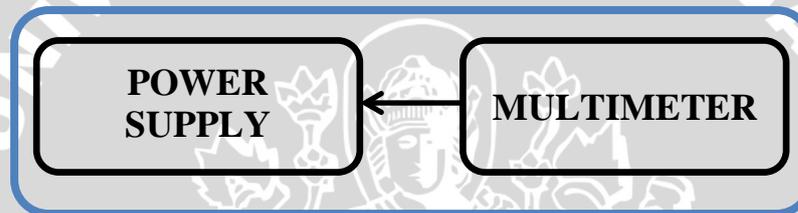
1. pengujian catu daya

2. pengujian perangkat LCD
3. pengujian data sensor Ultrasonik PING)))
4. pengujian rangkaian *driver* motor
5. pengujian secara keseluruhan sistem

5.1 pengujian Catu Daya (power supply)

Pengujian ini dilakukan dengan tujuan untuk mengetahui nilai catu daya apakah sesuai dengan spesifikasi yang di butuhkan sistem.

Langkah-langkah pengujian dilakukan dengan bantuan multimeter standard yang di hubungkan langsung dengan *Power Supply*. Untuk blok diagram pengujian catu daya ditunjukkan pada Gambar 5.1 dan hasil pengujian pada Gambar 5.2.



Gambar 5.1 Diagram Blok Pengujian Catu Daya

Sumber: Pengujian 2015



Gambar 5.2 Hasil pengujian Catu daya

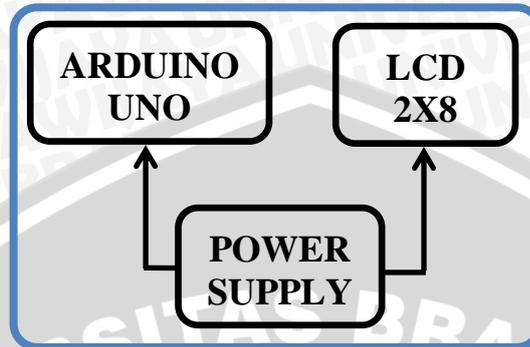
Sumber: Pengujian 2015

5.2 Pengujian Perangkat LCD

Pengujian ini dilakukan dengan tujuan untuk mengetahui nilai catu daya apakah sesuai dengan spesifikasi yang di butuhkan sistem. Perangkat LCD di

repository.ub.ac.id

gunakan untuk menampilkan pembacaan Data yang di lakukan oleh sistem, jadi memudahkan penguji untuk mendapatkan informasi. Untuk blok diagram pengujian LCD dapat di lihat pada gambar 5.3.



Gambar 5.3 Diagram Blok Pengujian LCD

Sumber: Pengujian 2015

5.2.1 Peralatan Pengujian Terdiri dari:

1. Kontroler Arduino UNO
2. Power Supply
3. LCD Board 2X8

5.2.2 Prosedur pengujian

Pada pengujian perangkat LCD, digunakan *Power Supply* yang di hubungkan ke perangkat kontroler yang sudah terpasang *LCD board*. *Power supply* di hidupkan dan nantinya akan menampilkan apakah LCD nyala sesuai yang telah di program. Dalam hal ini akan di tunjukkan dengan tulisan “Tes LCD” pada layar. Untuk hasil pengujian dapat di lihat pada Gambar 5.4.

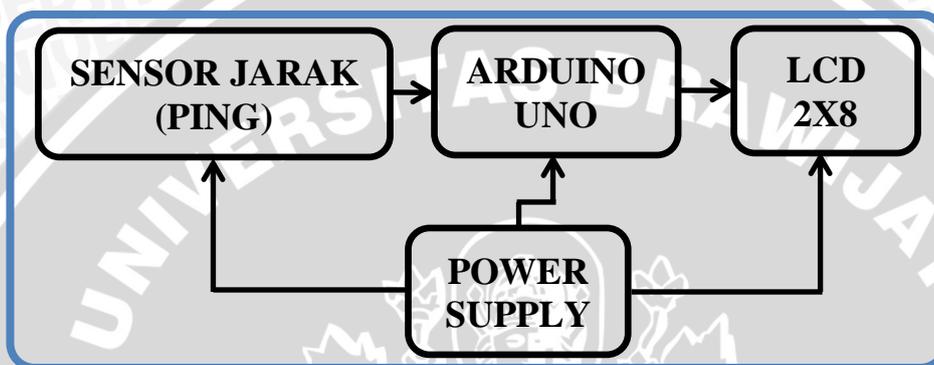


Gambar 5.4 Hasil Pengujian LCD

Sumber: Pengujian 2015

5.3 Pengujian data Sensor PING)))

Pengujian ini dilakukan dengan tujuan untuk mengetahui pembacaan sensor apakah sesuai dengan jarak sebenarnya. Langkah-langkah pengujian dilakukan dengan menghubungkan sensor ultrasonik, kontroler Utama Arduino UNO, *Power Supply* dan LCD. Pada Gambar 5.5 ditunjukkan blok diagram pada pengujian sensor Ping.



Gambar 5.5 Diagram Blok Pengujian Sensor Ping

Sumber: Pengujian, 2015

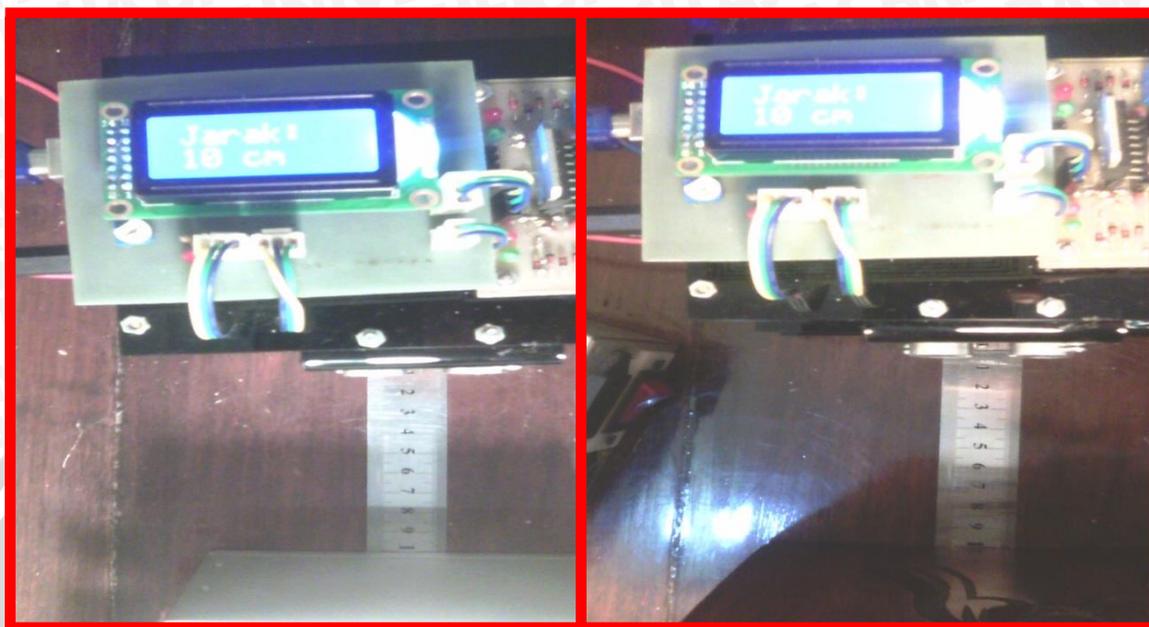
5.3.1 Peralatan pengujian terdiri dari:

1. Laptop
2. Arduino UNO
3. Sensor PING)))
4. Meteran/penggaris Panjang
5. *Power Supply*

5.3.2 Prosedur pengujian

Pada pengujian berikut ini, objek (penghalang) pada lintasan lurus diletakkan sejajar didepan sensor PING))). Jarak objek dengan sensor di ubah-ubah setiap 10cm dengan menggunakan penggaris panjang. Perangkat *timer 1* pada kontroler pengatur sensor PING))) digunakan untuk menghitung lamanya waktu aktif sinyal pantulan dari sensor PING))). Data berupa lama waktu dalam milisekon lalu jarak terbaca

dalam centimeter yang akan di tampilkan pada LCD. Gambar hasil pengujian bisa di lihat pada Gambar 5.6.



Gambar 5.6 Gambar hasil Pengujian Sensor PING))

Sumber: Pengujian, 2015

5.3.3 Hasil pengujian

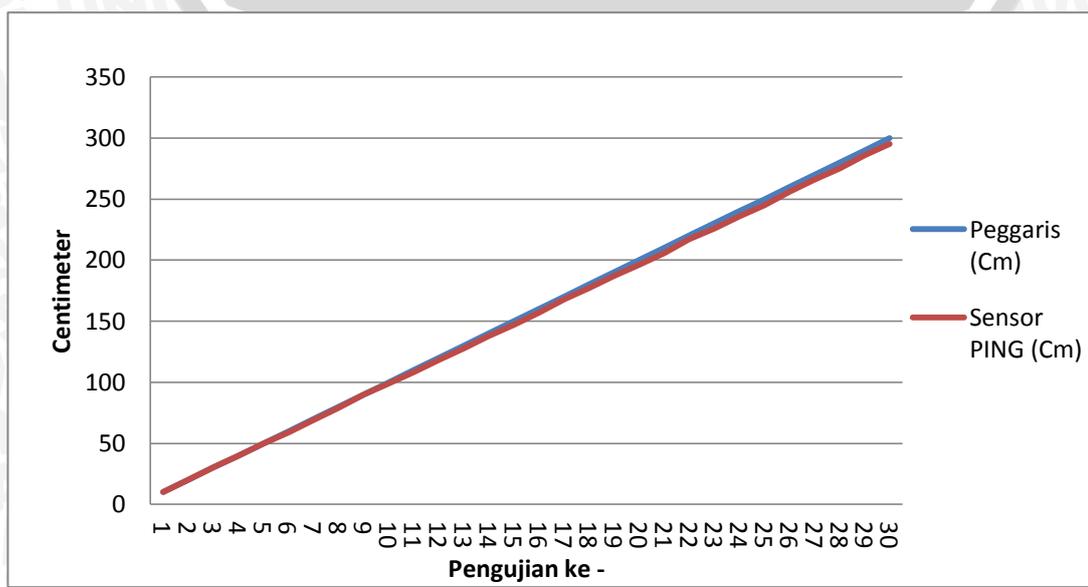
Hasil pengujian yang di peroleh dari pembacaan jarak sensor PING)) dengan jarak sebenarnya dalam beberapa kali pengambilan data dapat dilihat pada Tabel 5.1 dan grafik respon dapat di lihat pada Gambar 5.7 dan 5.8.

Tabel 5.1 Tabel hasil pengujian Sensor PING))

Sumber: Pengujian, 2015

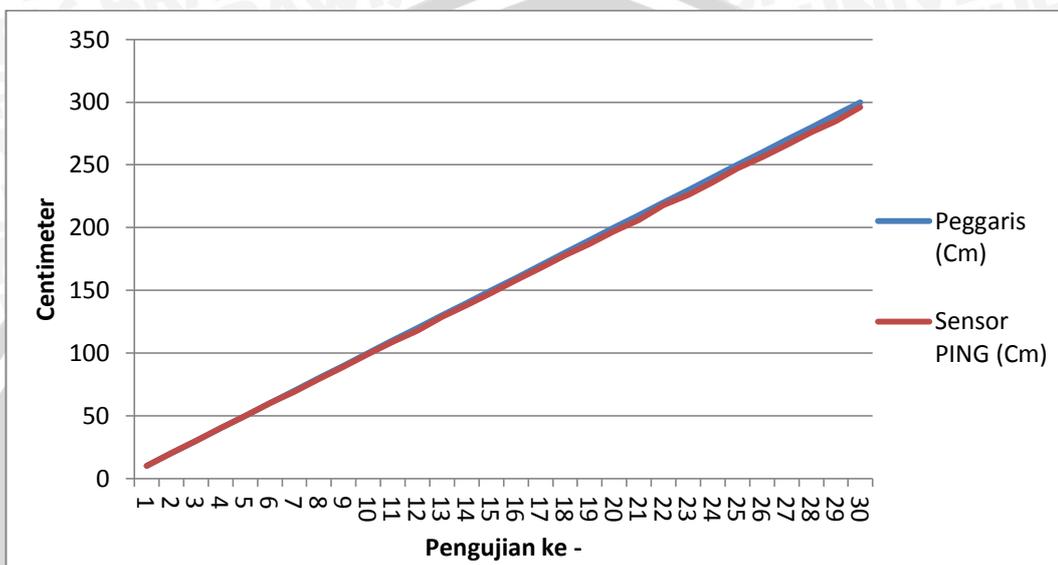
No	Penggaris	JARAK (Cm)					
		Sensor Jarak (PING)					
		Besi	Error (Besi)	% Error (Besi)	Kayu	Error (Kayu)	% Error (Kayu)
1	10	10	0	0%	10	0	0%
2	20	20	0	0%	20	0	0%
3	30	30	0	0%	30	0	0%
4	40	40	0	0%	40	0	0%

5	50	50	0	0%	50	0	0%
6	60	59	1	1%	60	0	0%
7	70	69	1	1%	69	1	1%
8	80	79	1	1%	79	1	1%
9	90	90	0	0%	89	1	1%
10	100	99	1	1%	99	1	1%
11	110	108	2	2%	109	1	1%
12	120	118	2	2%	118	2	2%
13	130	128	2	2%	129	1	1%
14	140	138	2	2%	138	2	2%
15	150	147	3	3%	148	2	2%
16	160	157	3	3%	158	2	2%
17	170	168	2	2%	168	2	2%
18	180	177	3	3%	178	2	2%
19	190	187	3	3%	187	3	3%
20	200	196	4	4%	197	3	3%
21	210	206	4	4%	206	4	4%
22	220	217	3	3%	218	2	2%
23	230	226	4	4%	226	4	4%
24	240	236	4	4%	236	4	4%
25	250	245	5	5%	247	3	3%
26	260	256	4	4%	256	4	4%
27	270	266	4	4%	266	4	4%
28	280	275	5	5%	276	4	4%
29	290	286	4	4%	285	5	5%
30	300	295	5	5%	296	4	4%
RATA-RATA			2.40	2%		2.07	2%



Gambar 5.7 Grafik hasil pengujian Sensor PING))) dengan Besi

Sumber: Pengujian, 2015

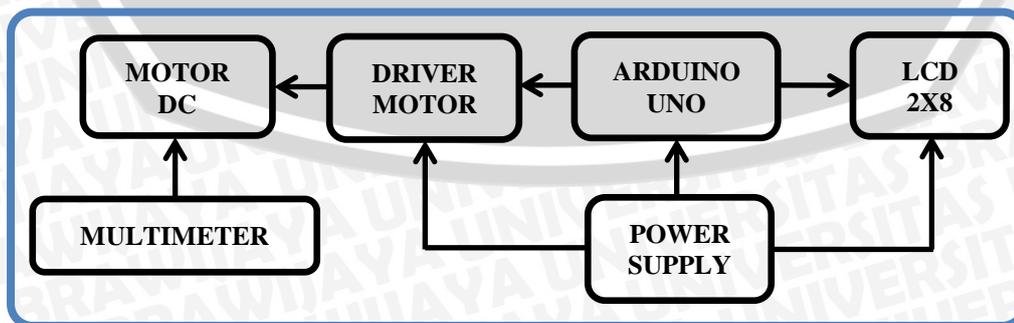


Gambar 5.8 Grafik hasil pengujian Sensor PING))) dengan Kayu

Sumber: Pengujian, 2015

5.4 Pengujian *Driver* Motor

Pengujian ini bertujuan untuk melihat apakah *supply* tegangan yang di berikan *driver* motor sesuai dengan yang di inginkan. Langkah pengujian dilakukan seperti pada diagram blok pengujian *driver* Motor yang di tunjukkan pada Gambar 5.9.



Gambar 5.9 Diagram Blok Pengujian *Driver* Motor

Sumber: Pengujian, 2015



5.4.1 Peralatan yang digunakan

1. Laptop
2. Kontroler Arduino UNO
3. *Multimeter*
4. Motor DC
5. *Driver Motor*
6. *Power Supply*

5.4.2 Prosedur pengujian

Prosedur pengujian dilakukan dengan memberi catu daya pada perangkat yang sudah di hubungkan dengan *Driver Motor*. Kontroler akan memberi perintah untuk mengatur prosentasi PWM tertentu sesuai yang telah di tentukan. Pengujian PWM diberikan dari skala 0-255 dan pengambilan data pada setiap interval 5. Pengujian dilakukan dua kali, yaitu dengan beban dan tanpa beban. Hasil pengujian akan di tampilkan langsung pada LCD.

5.4.3 Hasil Pengujian

Hasil pengujian *Driver motor* pada Prosentase PWM tertentu yang diperoleh melalui beberapa kali percobaan ditunjukkan dalam Tabel 5.2 dan grafik respon ditunjukkan pada Gambar 5.10.

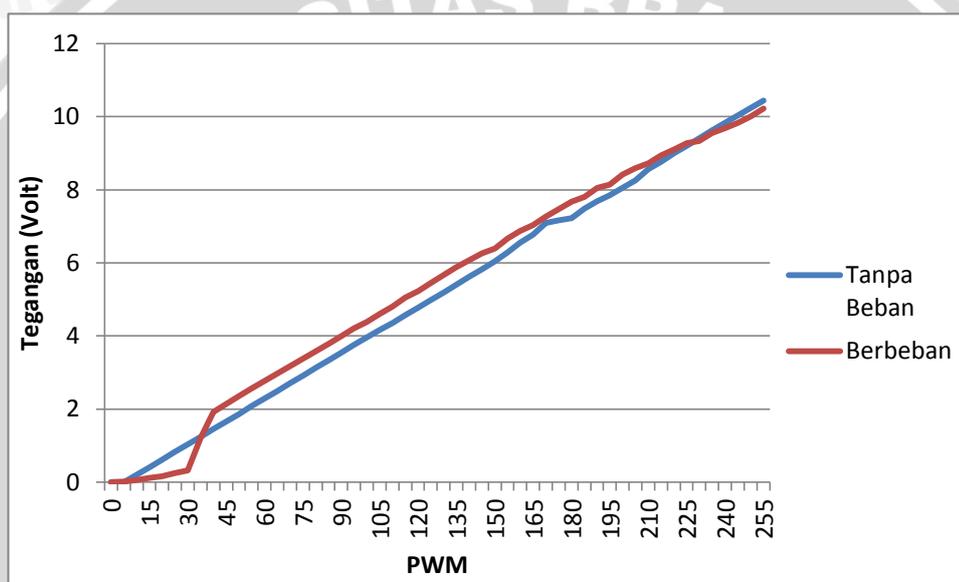
Tabel 5.2 Hasil pengujian *Driver Motor*
(Sumber: Pengujian, 2015)

NO	PWM	Tegangan (Volt)	
		Tanpa Beban	Berbeban
1	0	0	0
2	5	0.006	0.014
3	10	0.2	0.057
4	15	0.412	0.112
5	20	0.618	0.166
6	25	0.826	0.243
7	30	1.035	0.324



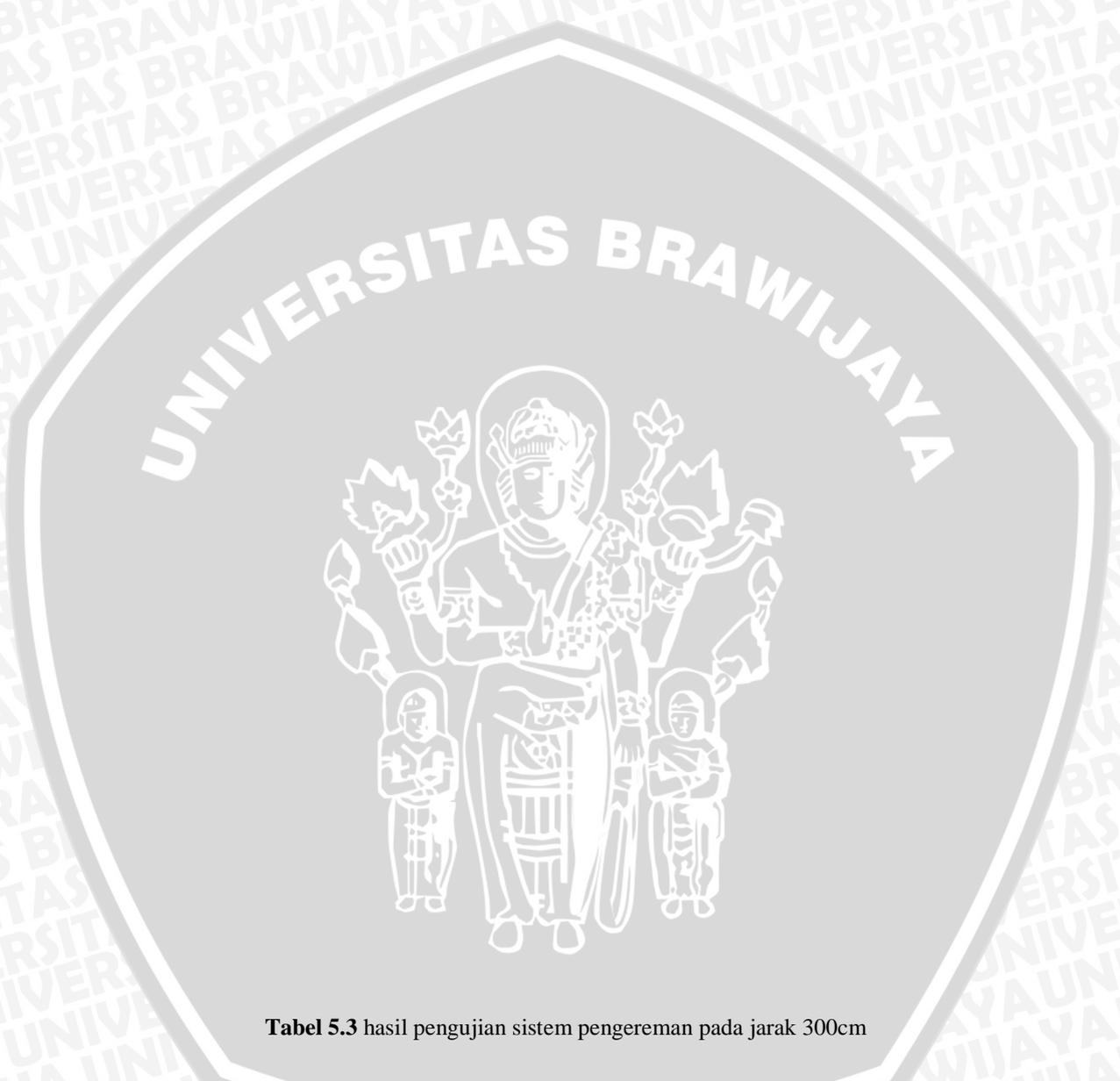
8	35	1.243	1.208
9	40	1.452	1.917
10	45	1.66	2.147
11	50	1.869	2.361
12	55	2.079	2.57
13	60	2.287	2.773
14	65	2.497	2.978
15	70	2.706	3.177
16	75	2.916	3.383
17	80	3.124	3.582
18	85	3.333	3.789
19	90	3.542	3.988
20	95	3.751	4.21
21	100	3.959	4.39
22	105	4.17	4.6
23	110	4.36	4.81
24	115	4.57	5.06
25	120	4.78	5.23
26	125	4.98	5.44
27	130	5.19	5.67
28	135	5.4	5.88
29	140	5.62	6.08
30	145	5.82	6.26
31	150	6.04	6.39
32	155	6.29	6.67
33	160	6.56	6.88
34	165	6.77	7.03
35	170	7.09	7.27
36	175	7.17	7.47
37	180	7.23	7.67
38	185	7.49	7.81
39	190	7.69	8.05
40	195	7.85	8.14
41	200	8.05	8.42
42	205	8.26	8.59
43	210	8.57	8.72

44	215	8.77	8.94
45	220	9	9.1
46	225	9.2	9.27
47	230	9.41	9.34
48	235	9.62	9.55
49	240	9.83	9.69
50	245	10.04	9.83
51	250	10.24	10
52	255	10.44	10.23



Gambar 5.10 Grafik hasil Pengujian *Driver* Motor Berbeban dan tidak berbeban

Sumber: pengujian 2015



Tabel 5.3 hasil pengujian sistem pengereman pada jarak 300cm

Jarak (cm) input		output
300	1228	1228
298,75	1219	1219
297,5	1214	1214
297,5	1214	1214
297,5	1214	1214

296,25	1210	1210
296,25	1210	1210
295	1205	1326
296,25	1210	1326
296,25	1210	1326
296,25	1210	1552



297,5	1214	1552
296,25	1210	1350
296,25	1210	1350
296,25	1210	1350
297,5	1214	1350
297,5	1214	1146
296,25	1210	1146
296,25	1210	1146
297,5	1214	1086
297,5	1214	1086
296,25	1210	1086
297,5	1214	1086
297,5	1214	1146
296,25	1210	1146
296,25	1210	1146
297,5	1214	1146
297,5	1214	1212
296,25	1210	1212
297,5	1214	1212
296,25	1210	1236
296,25	1210	1236
297,5	1214	1236
297,5	1214	1236
297,5	1214	1238
297,5	1214	1238
297,5	1214	1238
297,5	1214	1238
297,5	1214	1238
297,5	1214	1238
Jarak (cm)	Input (RPM)	Output (RPM)
297,5	1214	1238
297,5	1214	1236
297,5	1214	1236
297,5	1214	1236
297,5	1214	1236

296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
297,5	1214	1238
296,25	1210	1238
296,25	1210	1238
297,5	1214	1238
296,25	1210	1238
296,25	1210	1238
297,5	1214	1238
296,25	1210	1238
297,5	1214	1238
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
297,5	1214	1238
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
297,5	1214	1238
296,25	1210	1240
296,25	1210	1240
297,5	1214	1240
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
296,25	1210	1238
293,75	1201	1238
Jarak (cm)	Input	Output



	(RPM)	(RPM)
287,5	1178	1238
283,75	1164	1238
280	1150	1204
277,5	1141	1204
275	1132	1204
272,5	1122	1124
268,75	1109	1124
266,25	1099	1124
262,5	1086	1124
258,75	1072	1084
256,25	1063	1084
253,75	1053	1084
251,25	1044	1084
247,5	1030	1064
245	1021	1064
241,25	1007	1064
238,75	998	1028
233,75	980	1028
228,75	961	1028
227,5	957	1028
225	948	974
222,5	938	974
218,75	925	974
216,25	915	974
212,5	902	924
210	892	924
206,25	879	924
202,5	865	924
201,25	860	882
200	856	882
197,5	846	882
195	837	844
190	819	844
187,5	810	844

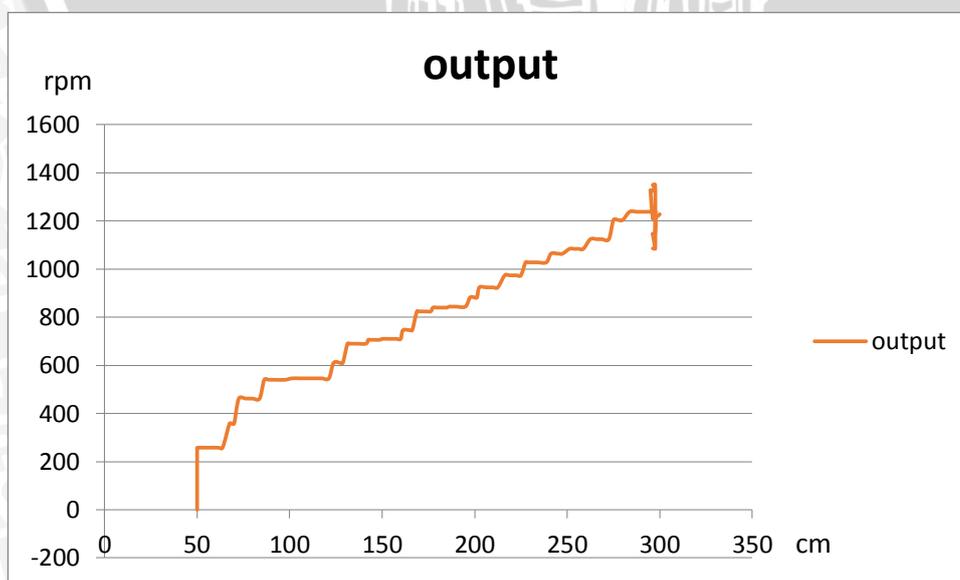
186,25	805	844
185	800	840
183,75	796	840
180	782	840
177,5	773	840
176,25	768	824
173,75	759	824
Jarak (cm)	Input (RPM)	Output (RPM)
168,75	741	824
166,25	731	746
165	727	746
161,25	713	746
160	708	710
157,5	699	710
153,75	685	710
150	672	710
148,75	667	706
146,25	658	706
143,75	649	706
142,5	644	706
141,25	639	690
136,25	621	690
132,5	607	690
131,25	603	690
128,75	593	610
127,5	589	610
123,75	575	610
121,25	566	546
117,5	552	546
115	543	546
111,25	529	546
110	524	546
107,5	515	546
105	506	546

101,25	492	546
97,5	478	540
92,5	460	540
88,75	446	540
86,25	437	540
83,75	428	462
80	414	462
76,25	400	462

72,5	386	462
70	377	358
68,75	373	358
67,5	368	358
63,75	354	258
61,25	0	258
50	0	258

Jarak (cm)	Input (RPM)	Output (RPM)
50	0	80
50	0	80
50	0	80
50	0	80
50	0	0
50	0	0
50	0	0
50	0	0
50	0	0
50	0	0
50	0	0
50	0	0

Grafik dari hasil pengujian pada Tabel 5.3 ditunjukkan pada Gambar 5.11.



Gambar 5.11 Grafik respon kerja sistem secara keseluruhan

Pada Gambar 5.11 grafik dengan menggunakan K_p , K_i dan K_d yang paling tepat, $K_p=0,02$ $K_i=0$ dan $K_d=0,054$ dan dari data hasil percobaan pada Tabel 5.3 didapatkan nilai:

$$\text{Overshoot maks} = \frac{(\text{nilai output maks} - \text{setpoint})}{\text{setpoint}} \times 100\%$$

$$= \frac{(1350 - 1214)}{1214} \times 100\%$$

$$= 11,2\%$$

$$\text{error} = (\text{nilai output} - \text{setpoint}) \times 100\%$$

$$\text{error rata-rata} = \frac{\text{jumlah prosentase error total tiap data}}{\text{banyaknya data}}$$

$$= \frac{47,72}{162} \times 100\%$$

$$= 0,029457 \times 100\% = 2,94\%$$

Nilai *prosentase overshoot maks* sistem 11,2% dihitung dari nilai keluaran maksimum sistem, dengan *error rata-rata* 2,94% didapat dari jumlah total *error* pada tiap data pada Tabel 5.3.

BAB VI

KESIMPULAN

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kesimpulan yang dapat diambil dalam pembuatan penelitian ini adalah sebagai berikut :

1. Alat yang dirancang sudah sesuai dengan yang diharapkan, di mana Sistem dapat melakukan pegeraman secara elektris dibawah 300cm dan akan berhenti pada jarak kurang dari 50cm dengan rata-rata *error* 2,94% dan nilai *overshoot* 11,2%.
2. Dengan menggunakan metode *hand tuning* untuk menentukan nilai parameter kontroler PID, maka didapatkan parameter yang terbaik yaitu $K_p = 0.02$, $K_i = 0$ dan $K_d = 0,054$. Nilai $K_i = 0$ dikarenakan motor DC yang digunakan tidak 100% linier, pada rpm di atas 300 baru motor bejalan linier.

6.2 Saran

Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan. Untuk memperbaiki kinerja alat dan pengembangan lebih lanjut Disarankan menggunakan Motor DC yang linier secara keseluruhan untuk mendapat kan respon motor yang cepat dan tepat.

Daftar Pustaka

- A.E. Fitzgerald, Kingsley Charles, Umans D Stephen, dan Achyanto Djoko. *Mesinmesin listrik-edisi keempat*. Jakarta. Erlangga.1997.
- Amstrom, K. J., & Hagglund, Tore. 1995 *PID Controllers: Theory, Design and Tuning*. Instrument Society of America: Research Triangle Park.
- Baharuddin, Rhiza S.Sadjad, Tola Muhammad. 2012. Sistem pengendalian kecepatan motor DC berbasis *pwm (pulse width modulation)*. Sulawesi Tenggara. Universitas Hasanudin.
- Barley,Eres.2011. Sistem Pengaturan Kecepatan Putaran Motor DC Pada Konveyor Bertingkat Menggunakan Kontroler PID Berbasis PLC.Skripsi. tidak dipublikasikan. Malang: Universitas Brawijaya.
- Firmansyah, E. *Pengukur Jarak Dengan Gelombang Ultrasonik Memanfaatkan Mikrokontroler 68HC11A1FN*, Tugas Akhir, Universitas Gadjah Mada,Yogyakarta. 2001.
- Laboratorium Sistem Kontrol. 2010. *Praktikum Sistem Kontrol*. Malang: Universitas Brawijaya.
- Parallax, 2008. *PING))) Ultrasonic Distance Sensor*. California: Parallax.
- Sulasno dan agus prayitno, Thomas. *Teknik sistem control-Edisi pertama*.Yogyakarta. Graha ilmu. 2006.
- Sudjadi, *Teori dan Aplikasi Mikrokontroler*, Penerbit Graha Ilmu, Yogyakarta, 2005.
- Setyanoveka, Deaz Achmedo Giovanni, 2014. Sistem pengendali perlambatan kecepatan motor pada robot *line follower* dengan sensor Ultrasonik. Malang: Skripsi Jurusan Teknik Elektro FT-UB.

-----<http://www.dasar-elektronika.web.id>

-----<http://www.atmel.com>

-----<http://www.parallax.com>

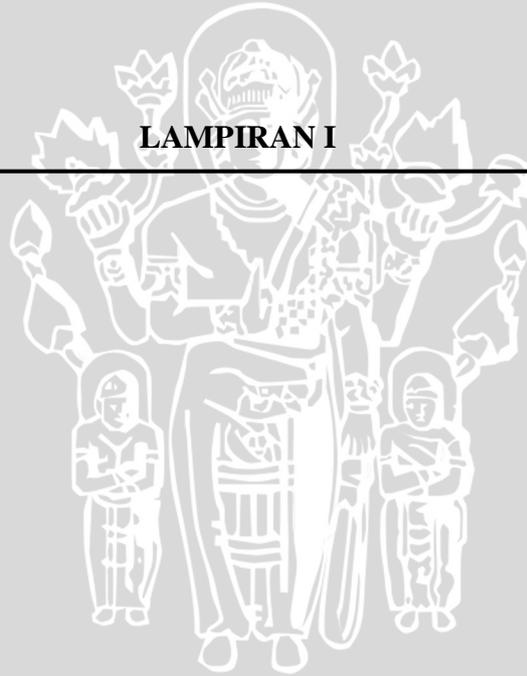
-----<http://www.elektro.studentjournal.ub.ac.id>

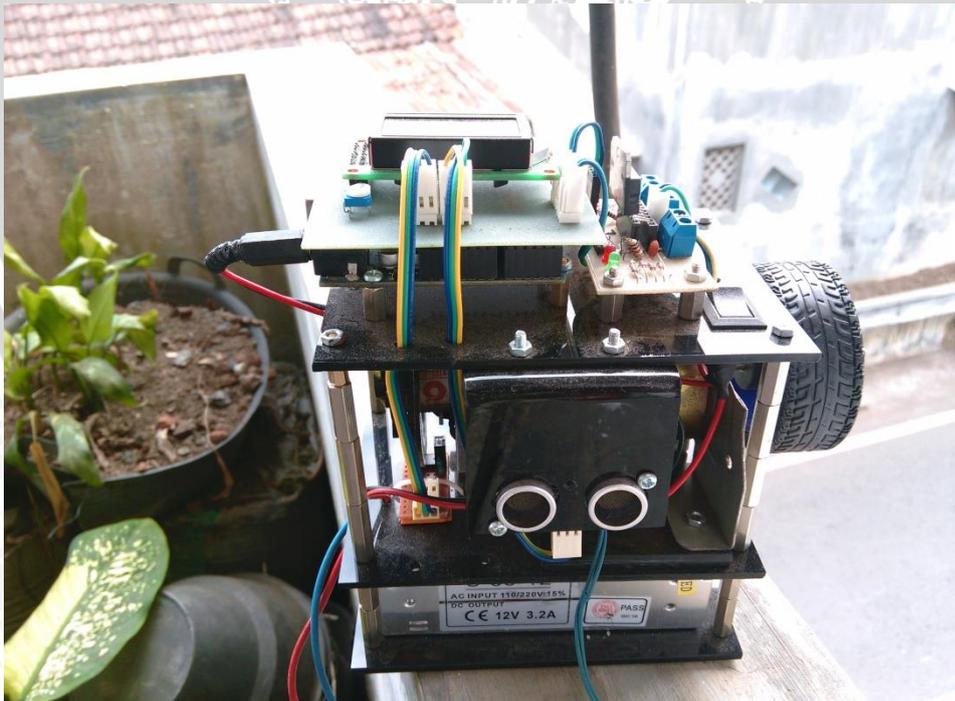
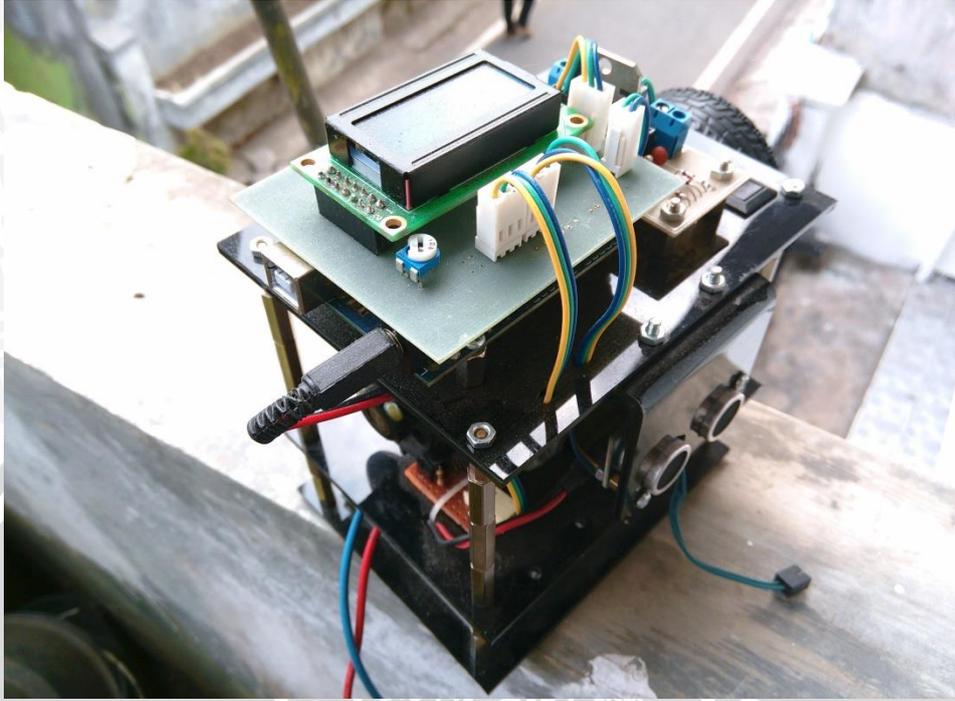
----- <https://www.arduino.cc>

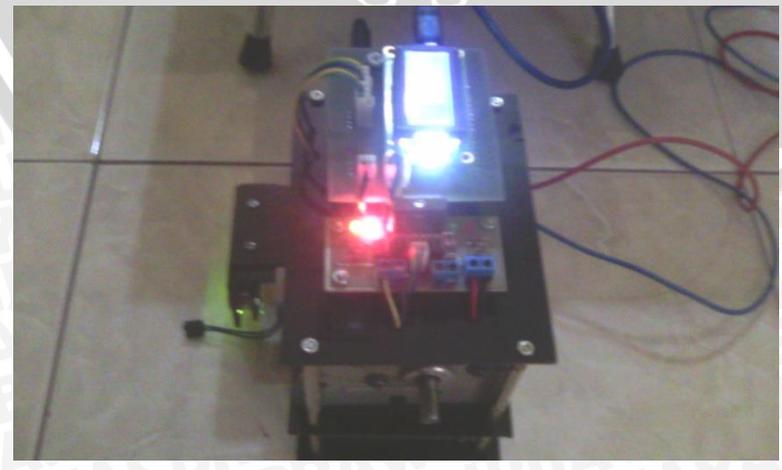
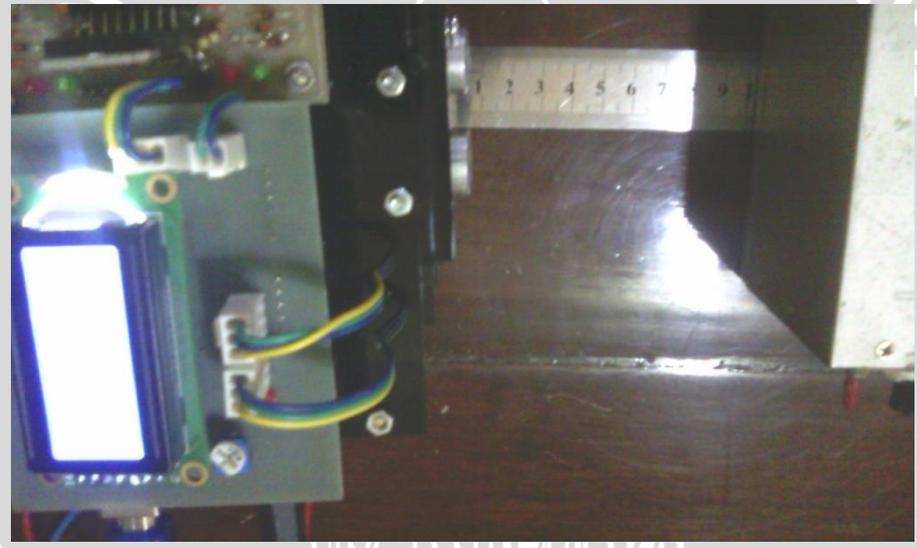
UNIVERSITAS BRAWIJAYA

LAMPIRAN I

Foto alat













LAMPIRAN II

Listing Program



- **Listing Program PID**

```
//-----PID PROGRAM-----
#include "Timer.h"
#include <LiquidCrystal.h>

Timer t;
int pwmMotor, lastError;
int error = 0;
double Output;
const int pingPin = 7;
double kp, ki, kd;
float pulsa;
float Ts=100;
int kecepatan;
int motorPinPWM = 11;
int motorCW = 8;
int motorCCW = 10;
int n, setpoint;
unsigned long lastTime;
long duration, cm;
LiquidCrystal lcd (12, 9, 3, 6, 4, 5);

void setup()
{
```



```

kp = 0.02;//0.02
ki = 0;//0
kd = 0.054;//0.054

lcd.begin(8, 2);
Serial.begin(9600);
Serial.print("PWM");
Serial.print("\t");
Serial.println("Speed");
pinMode(motorPinPWM, OUTPUT);
pinMode(motorCW, OUTPUT);
pinMode(motorCCW, OUTPUT);
digitalWrite(motorCW, HIGH);
//digitalWrite(motorPinPWM, HIGH);
n=0;

//inisialisasi timer

noInterrupts(); // disable interrupts
TCCR1A = 0;
TCCR1B = 0;
TCNT1 = 0;
OCR1A = 62499; // compare match register 16MHz/256/50Hz/1000ms
TCCR1B |= (1 << WGM12); // CTC mode
TCCR1B |= (1 << CS12); // 256 prescaler
TIMSK1 |= (1 << OCIE1A); // enable timer compare interrupt
attachInterrupt(0, hitung_pulsa, FALLING);
interrupts(); // enable all interrupts

Serial.begin(9600);
}

```

```
ISR(TIMER1_COMPA_vect) // timer compare interrupt service routine
```

```
{
    kecepatan = (pulsa/30) * 60;
    pulsa = 0;
}
```

```
long microsecondsToCentimeters(long microseconds)
```

```
{
    return microseconds / 29 / 2;
}
```

```
void hitung_pulsa()
```

```
{
    pulsa++;
}
```

```
void compute()
```

```
{
    digitalWrite(motorCW, HIGH);digitalWrite(motorCCW, LOW);
    unsigned long now = millis();
    double timeChange = (double)(now-lastTime);

    error = setpoint - kecepatan;
    Output = (kp * error) + ((ki*timeChange) *(error + lastError)) +
    ((kd/timeChange)*(error - lastError));
```

```
if(Output > 1)pwmMotor=pwmMotor + Output;
```

```
else if (Output < -1)pwmMotor= pwmMotor + Output;
```

```
else pwmMotor = pwmMotor;
```

```
if(pwmMotor<0)pwmMotor=0;
```

```
else if(pwmMotor>255)pwmMotor=255;
```

```
lastError = error;
```

```
lastTime = now;
}

void Tune_Setpoint()
{
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);

  cm = microsecondsToCentimeters(duration);
  setpoint = map(cm, 50, 300, 350, 1500);
  if(setpoint<=350)
  setpoint=0;
  digitalWrite(motorCW, LOW);
  digitalWrite(motorCCW, HIGH);
}

void loop()
{
  Tune_Setpoint();
  compute();
  analogWrite(motorPinPWM, pwmMotor);

  lcd.setCursor(0,0);
  lcd.print("S:");
  lcd.print(setpoint);

  lcd.setCursor(0,1);
```

```

lcd.print("K:");
lcd.print(kecepatan);
delay(250);//250
lcd.clear();

Serial.print(cm);
Serial.print("\t");
Serial.print(setpoint);
Serial.print("\t");
Serial.print(kecepatan);
Serial.print("\t");
Serial.println(pwmMotor);
}

```

- **Pengujian LCD**

```

// The circuit:
// * LCD RS pin to digital pin 12
// * LCD Enable pin to digital pin 11
// * LCD D4 pin to digital pin 3
// * LCD D5 pin to digital pin 6
// * LCD D6 pin to digital pin 4
// * LCD D7 pin to digital pin 5
// * LCD R/W pin to ground
// * LCD VSS pin to ground
// * LCD VCC pin to 5V

```

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(12, 11, 3, 6, 4, 5);
```

```
void setup()
```

```

{
  lcd.begin(8,2);
}

void loop()
{
  lcd.setCursor(0,0);
  lcd.print("TES LCD");
}

```

- **PENGUJIAN KECEPATAN**

```

//pin baca sensor rotary ==> pin2
#include <LiquidCrystal.h>
#include ppr 30
//#include "Timer.h"

LiquidCrystal lcd(12, 11, 3, 6, 4, 5);

//Timer t;
float pulsa;
int kecepatan;

void setup()
{
  Serial.begin(9600);
  lcd.begin(8, 2);
  pinMode(9, OUTPUT);
  pinMode(10,OUTPUT);

  //inisialisasi timer

  noInterrupts(); // disable interrupts
  TCCR1A = 0;

```



```

TCCR1B = 0;
TCNT1 = 0;
OCR1A = 62499; // compare match register 16MHz/256/50Hz/1000ms
TCCR1B |= (1 << WGM12); // CTC mode
TCCR1B |= (1 << CS12); // 256 prescaler
TIMSK1 |= (1 << OCIE1A); // enable timer compare interrupt
attachInterrupt(0, hitung_pulsa, FALLING);
interrupts(); // enable all interrupts

digitalWrite(9, HIGH);
digitalWrite(10, HIGH);
}

ISR(TIMER1_COMPA_vect) // timer compare interrupt service routine
{
    kecepatan = (pulsa/30) * 60;
    pulsa = 0;
}

void hitung_pulsa()
{
    pulsa++;
}

void loop()
{
    lcd.setCursor(0,0);
    lcd.print(kecepatan);
}

```

- **PENGUJIAN SENSOR PING))**

```
#include <LiquidCrystal.h>
```

```
const int pingPin = 7;
int setpoint;

LiquidCrystal lcd(12, 9, 3, 6, 4, 5);

void setup()
{
  lcd.begin(8,2);
  lcd.setCursor(0,0);
  lcd.print("UJI PING");
  delay(1000);
  Serial.begin(9600);
}

void loop()
{
  long duration, cm;

  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);

  cm = microsecondsToCentimeters(duration);
  setpoint = map(cm, 50, 300, 0, 1500);
  if(setpoint<=0)setpoint=0;
```

```

lcd.setCursor(0,0);
lcd.print("J:");
lcd.print(cm);
lcd.setCursor(0,1);
lcd.print("S:");
lcd.print(setpoint);
Serial.print(cm);
Serial.print("\t");
Serial.println(setpoint);
delay(250);
lcd.clear();
}

```

```

long microsecondsToCentimeters(long microseconds)

```

```

{
  return microseconds / 29 / 2;
}

```

- **PENGUJIAN PWM**

```

#include <LiquidCrystal.h>

```

```

LiquidCrystal lcd(12, 11, 3, 6, 4, 5);

```

```

int motorPinPWM = 9;

```

```

int motorCW = 8;

```

```

int motorCCW = 10;

```

```

int n;

```

```

void setup()

```

```

{
  lcd.begin(8, 2);
  pinMode(motorPinPWM, OUTPUT);
  pinMode(motorCW, OUTPUT);
}

```

```
pinMode(motorCCW, OUTPUT);  
lcd.setCursor(0,0);  
lcd.print("UJI PWM");  
delay(1000);  
lcd.clear();  
digitalWrite(motorCCW, HIGH);  
n=0;  
}
```

```
void loop()
```

```
{  
while(n<=255)  
{  
n=n;  
analogWrite(motorPinPWM,n);  
lcd.setCursor(0,0);  
lcd.print("PWM:");  
lcd.setCursor(0,1);  
lcd.print(n);  
delay(1500);  
lcd.clear();  
n=n+5;  
}  
}
```





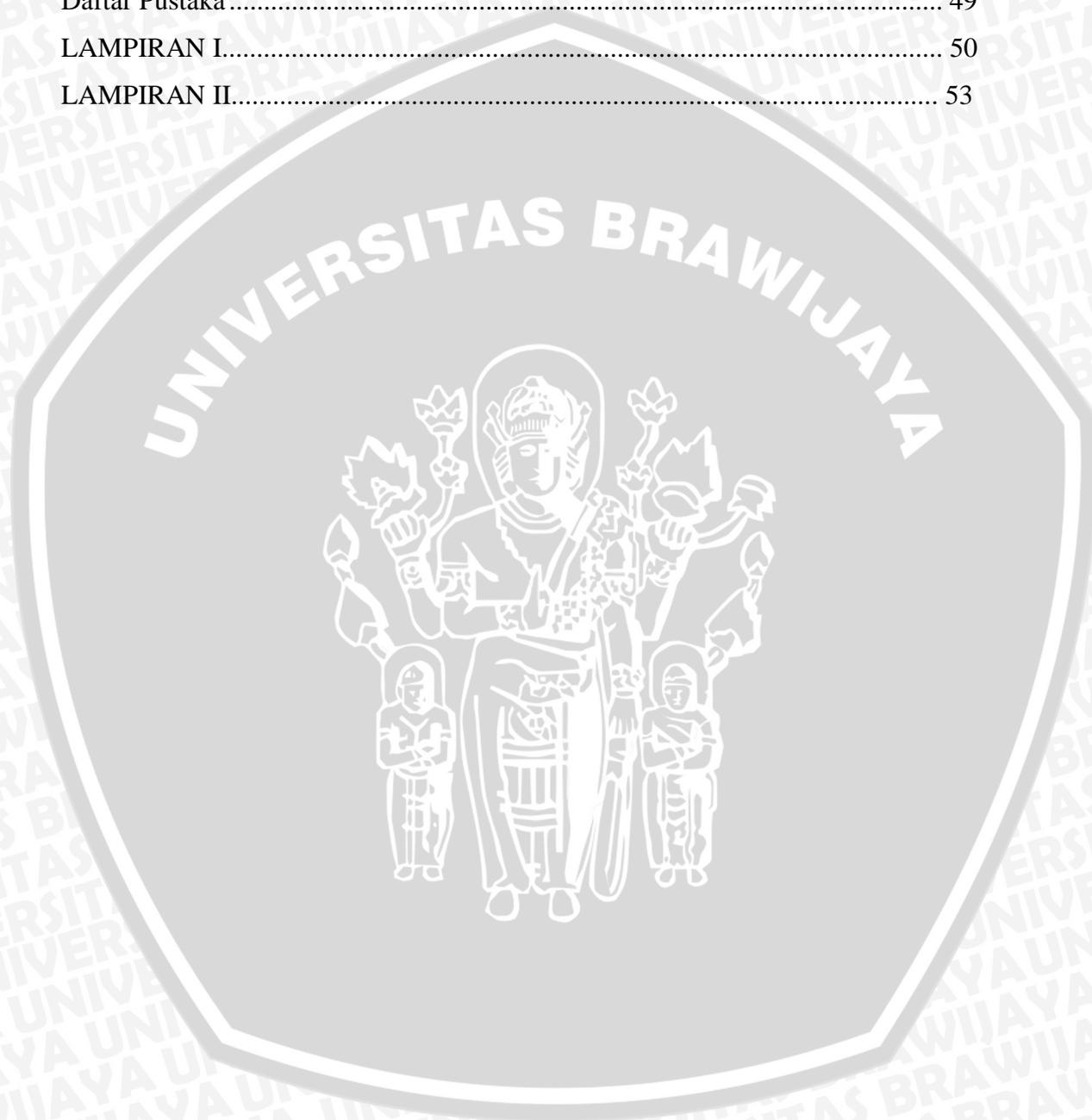
DAFTAR ISI

KATA PENGANTAR	i
ABSTRAK	iii
BAB I	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 TUJUAN.....	3
1.5 Sistematika Pembahasan	3
BAB II.....	4
TINJAUAN PUSTAKA.....	4
2.1 MOTOR DC	4
2.2 Pulse Width Modulation (PWM).....	7
2.3 Sensor Ultrasonic.....	9
2.3.1 Prinsip kerja sensor ultrasonik	11
2.4 <i>Driver</i> Motor	11
2.5 Arduino Uno.....	12
2.5.1 Circuit Board.....	Error! Bookmark not defined.
2.5.2 Catu Daya.....	14
2.5.3 Memory ATmega328	15
2.5.4 Input & Output Arduino UNO	17
2.5.5 Komunikasi Arduino UNO	18
2.5.6 Programming.....	18
2.5.7 Perangkat Lunak (Arduino IDE).....	17
2.5.8 Otomatis Software Reset.....	19
2.6 Kontroler Proporsional Integral Derivative (PID).....	20
2.6.1 Pengontrol proporsional	18
2.6.2 Pengontrol Integral.....	22
BAB III.....	24
METODE PENELITIAN.....	24



3.1	Diagram Alir Prinsip kerja Alat	24
3.2	Perancangan dan Pembuatan Alat	25
3.3	Perancangan Elektrik.....	25
3.4	Perancangan dan Pembuatan Perangkat Lunak (<i>Software</i>).....	26
3.5	Pengujian dan Analisis	26
3.6	Pengambilan Kesimpulan dan Saran	26
3.7	SISTEMATIKA PEMBAHASAN	26
BAB IV		28
PERANCANGAN DAN PEMBUATAN ALAT		28
4.1	BLOK DIAGRAM SISTEM.....	28
4.2	Prinsip Kerja Alat.....	29
4.3	Perancangan Perangkat Keras (<i>Hardware</i>)	29
4.3.1	Perancangan Mekanik Sistem	29
4.4	Perancangan Sistem Kerja.....	31
4.5	Perancangan Catu Daya Sistem.....	31
4.6	Perancangan Sensor Ultrasonik (PING)).....	32
4.7	Perancangan Perangkat Lunak	32
BAB V		33
PENGUJIAN DAN ANALISIS.....		33
5.1	pengujian Catu Daya (power supply)	34
5.2	Pengujian Perangkat LCD	34
5.2.1	Peralatan Pengujian Terdiri dari:	35
5.2.2	Prosedur pengujian.....	35
5.2.3	Hasil Pengujian	Error! Bookmark not defined.
5.3	Pengujian data Sensor (PING))	36
5.3.1	Peralatan pengujian terdiri dari:.....	36
5.3.2	Prosedur pengujian.....	36
5.3.3	Hasil pengujian.....	37
5.4	Pengujian <i>Driver</i> Motor	39
5.4.1	Peralatan yang digunakan	40
5.4.2	Prosedur pengujian.....	40
5.4.3	Hasil Pengujian	40

BAB VI.....	48
KESIMPULAN.....	48
6.1 Kesimpulan.....	48
6.2 Saran.....	48
Daftar Pustaka.....	49
LAMPIRAN I.....	50
LAMPIRAN II.....	53



DAFTAR GAMBAR

Gambar 2.1 Motor DC sederhana..... 4

Gambar 2.2 Amature terhubung ke sumber dc Es..... 6

Gambar 2.3 Amature terhubung Rangkaian terbalik..... 6

Gambar 2.4 Gelombang kotak $f(t)$ yang ideal dengan periode T 7

Gambar 2.5 Sinyal PWM dengan Variasi Duty Cycle 7

Gambar 2.6 Hubungan Nilai Duty Cycle dengan Tegangan Rata-rata 8

Gambar 2.7 Sensor PING))) /Ultrasonic 9

Gambar 2.8 Skema sensor ultrasonik 9

Gambar 2.6 Konfigurasi Pin IC L298 12

Gambar 2.9 Board Arduino UNO..... 13

Gambar 2.10 Kabel Usb Board Arduino UNO..... 13

Gambar 2.11 Konfigurasi Pin Atmega 328 15

Gambar 2.9 Tampilan Framework Arduino UNO..... 19

Gambar 2.10 Diagram blok kontroler proporsional 21

Gambar 2.11 Proportional band dari pengontrol proporsional tergantung pada penguatan. 19

Gambar 2.12 Kurva sinyal kesalahan $e(t)$ terhadap t pada pembangkit kesalahan nol..... 20

Gambar 3.1 Diagram Blok sistem 24

Gambar 3.2 Diagram Alir Sistem Kerja Alat 25

Gambar 4.1. Blok Diagram Sistem..... 28

Gambar 4.2 Desain 3D Mekanik alat keseluruhan 28

Gambar 4.3 Gambar keseluruhan alat 28

Gambar 4.4 Skema catu daya sistem 32

Gambar 4.5 sensor PING)))..... **Error! Bookmark not defined.**

Gambar 4.6 Diagram alir sistem Perangkat lunak 33

Gambar 5.1 Diagram Blok Pengujian Catu Daya 34

Gambar 5.2 Hasil pengujian Catu daya 34

Gambar 5.3 Diagram Blok Pengujian LCD..... 35

Gambar 5.4 Hasil Pengujian LCD 35



Gambar 5.5 Diagram Blok Pengujian Sensor Ping 36

Gambar 5.6 Gambar hasil Pengujian Sensor PING)))..... 37

Gambar 5.7 Grafik hasil pengujian Sensor PING))) dengan Besi..... 39

Gambar 5.8 Grafik hasil pengujian Sensor PING))) dengan Kayu 39

Gambar 5.9 Diagram Blok Pengujian *Driver* Motor..... 39

Gambar 5.10 Grafik hasil Pengujian *Driver* Motor Berbeban dan tidak berbeban 42

Gambar 5.11 Grafik respon kerja sistem secara keseluruhan..... 47

DAFTAR TABEL

Tabel 5.1 Tabel hasil pengujian Sensor PING))) 37

Tabel 5.2 Hasil pengujian *Driver* Motor 40

Tabel 5.3 hasil pengujian sistem pengereman pada jarak 300cm 43

