PENGANTAR

Puji syukur Alhamdulillah penulis panjatkan kepada Allah SWT berkat limpahan rahmat dan hidayah-Nya penulis dapat menyelesaikan Laporan skripsi yang berjudul "Sistem Kontrol Posisi Untuk Robot "Eco" Pada Kontes Robot ABU Indonesia Menggunakan Metode PID". Laporan ini dibuat dengan tujuan untuk memenuhi persyaratan memperoleh gelar Sarjana Teknik pada Jurusan Teknik Elektro Konsentrasi Teknik Kontrol Fakultas Teknik Universitas Brawijaya Malang.

Dalam proses penulisan Laporan skripsi ini tidak sedikit hambatan yang penulis hadapi. Penulis menyadari bahwa kelancaran mulai dari penyusunan awal penelitian, pada saat penelitian dan penulisan Laporan ini berkat bantuan, dorongan, dan bimbingan dari berbagai pihak baik secara langsung maupun tidak langsung. Untuk itu penulis menyampaikan terimakasih yang sebesar-besarnya kepada :

1. Tuhan Yang Maha Esa Allah SWT, atas limpahan rahmat dan hidayah-Nya sehingga penyusunan skripsi ini dapat diselesaikan.
2. Bapak, Ibu dan seluruh anggota keluarga, atas dukungan dan doa yang telah diberikan.
3. Yang terhormat Bapak M. Aziz Muslim, ST., MT., Ph.D. selaku ketua Jurusan Teknik Elektro Universitas Brawijaya.
4. Yang terhormat Bapak Ir. Purwanto, MT. Selaku Ketua Kelompok Dosen Konsentrasi Teknik Kontrol Jurusan Teknik Elektro Universitas Brawijaya.
5. Yang terhormat Bapak M. Aziz Muslim, ST., MT., Ph.D. selaku dosen pembimbing 1 yang selalu memberikan bimbingan, arahan, dan motivasi dalam penyusunan Skripsi ini.
6. Yang terhormat Bapak Goegoes Dwi Nusantoro, ST., MT. Selaku dosen pembimbing 2 yang selalu memberikan bimbingan, arahan, solusi, dan motivasi dalam penyusunan Skripsi ini.
7. Teman-teman Tim robot angkatan 2012 Roni, Agus, Firman, Septian, Brian Reza kawalta, Ricky, Sofyan andika yusuf, Anggi, wiwin, dan juli atas dukungan, bimbingan dan motivasinya.
8. Teman-teman seluruh anggota Tim Robot Universitas Brawijaya.
9. Teman-teman tim robot KRAI 2015-2016 Agus, Hasyim, Dicka, Arfai, rizky, gilang luih, gilang R, enggar, dan andy.
10. Teman-teman konsentrasi Teknik Kontrol Hanif, Beni, Hesa, Tio, Denis, Hilmy, Andri, Suro, dan lainnya yang tak bisa penulis sebutkan satu persatu.

11. Teman-teman d'viceroy yang telah banyak memotivasi.
12. Teman-teman bermain Rizqy amalia, enyak, itis, mar, dan mamat.
13. Teman-teman Teknik Elektro Angkatan 2012 yang telah memberikan semangat, dorongan, dan motivasinya.

Penulis berharap laporan ini dapat bermanfaat dan dapat dijadikan referensi dimasa yang akan datang. Penulis sadar bahwa laporan ini masih banyak kekurangan dan masih jauh dari kesempurnaan. Oleh karena itu kritik dan saran yang membangun penulis harapkan demi kesempurnaan laporan ini.

Malang, Agustus 2016

Penulis

RINGKASAN

Fikrul Jihad, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Agustus 2016, *Sistem Kontrol Posisi Untuk Robot "Eco" Pada Kontes Robot ABU Indonesia Menggunakan Metode PID* , Dosen Pembimbing: M. Aziz Muslim, S.T., M.T., Phd. Dan Goegoes Dwi Nusantoro, S.T., M.T.

Kontes Robot ABU Indonesia (KRAI) 2016 dengan tema "Clean Energy Recharging The World" mempunyai rule yang berbeda dari tahun-tahun sebelumnya. Mahasiswa diwajibkan membuat 2 robot yaitu robot Eco dan robot Hybrid. Masing-masing robot mempunyai tugas yang berbeda. Robot Eco bertugas membawa propeller dari start zone menuju ke wind turbine station. Robot eco hanya mempunyai 1 aktuator yang digunakan sebagai steering untuk mengkontrol posisi dari robot. Dalam penelitian ini robot eco yang dibuat mempunyai 2 motor DC yang digunakan untuk menggerakkan robot maju tanpa mengkontrol posisi dari robot dari robot Eco. Untuk mengkontrol posisi dari robot menggunakan kontroler PID. Untuk mencari parameter Kp, Ki, dan Kd menggunakan metode root locus. Hasil perhitungan dengan pole = -7,99 diperoleh nilai Kp=4,4698, Ki=10, dan Kd=0,2797. Untuk mengambil data kami menggunakan lintasan yang sama dengan Kontes Robot ABU Indonesia (KRAI). Lintasan ini dibagi menjadi 4 lintasan pengujian yaitu Lintasan Pengujian A, Lintasan Pengujian B, Lintasan Pengujian C, dan Lintasan Pengujian D. Pengujian pada lintasan B memperoleh hasil, jika menggunakan kontroler PID mempunyai tingkat keberhasilan 80% dan jika tidak menggunakan kontroler PID adalah 40%. Dengan simpangan rata-rata jika menggunakan kontroler PID adalah 13,05° dan ketika tidak menggunakan kontroler PID adalah 22,8°. Pengujian pada lintasan D memperoleh hasil, jika menggunakan kontroler PID mempunyai tingkat keberhasilan 80% dan jika tidak menggunakan kontroler PID adalah 20%. Dengan simpangan rata-rata jika menggunakan kontroler PID adalah 12,34° dan ketika tidak menggunakan kontroler PID adalah 22,03°.

Kata Kunci : PID, *Line follower*.

SUMMARY

Fikrul Jihad, *Department of Electrical Engineering, Faculty of Engineering University of Brawijaya, August 2016, Control System Position for "ECO"Robot in Kontes Robot ABU Indonesia Using PID controller, Academic Supervisor:*M. Aziz Muslim, S.T., M.T., Phd. *and* Goegoes Dwi Nusantoro, S.T., M.T.

Kontes Robot ABU Indonesia (KRAI) 2016 with theme "Clean Energy Recharging The World" have a different rule from the previous year. College students must make 2 different robot that is a Eco robot and a hybrid robot. Every robot have a different mission. Eco robot have a mission to bring a propeller from start zone to the wind turbine station. Robot only have one actuator for steering to control the position of robot. To control the position of robot a PID controller is employed. To find parameters Kp, Ki, and Kd using root locus method. The parameter's result with pole s = -7,99 was obtained the value of PID parameters, that is Kp=4,4695, Ki=10, Kd=0,2797. To take the data we used same track with Kontes Robot ABU Indonesia (KRAI). This track divided into 4 experiment track that is Experiment track A, Experiment track B, Experiment track C, and Experiment track D. Experiment from track B was obtained a result, by using PID controller have a succes rate of 80% and if without using PID controller is of 40%. With average deviation for PID controller is 13.05° and without using PID controller is 22.8°. Experiment from track D was obtained a result, by using PID controller have a succes rate of 80% and if without using PID controller is of 20%. With average deviation for PID controller is 12.34° and without using PID controller is 22.03°.

Keywords : PID, Line follower.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1. Latar Belakang

Kontes Robot ABU Indonesia (KRAI) adalah salah satu kategori yang di lombakan pada Kontes Robot Indonesia (KRI).Kontes Robot Indonesia adalah ajang tahunan yang diselenggarakan oleh Dikti.Tujuan dari perlombaan KRAI adalah mencari perwakilan dari Indonesia untuk mengikuti ABU Robocon. Rule yang berlaku di KRAI mengacu pada ABU Robocon. Rule yang dikeluarkan oleh ABU Robocon setiap tahun nya pasti berbeda tergantung dari Negara penyelenggara nya. Dimana tema ABU Robocon 2016 adalah Clean Energy Recharging the world dan untuk penyelenggaranya adalah Negara Thailand. Pada rule tahun ini setiap tim diharuskan membuat 2 robot. Robot pertama diperbolehkan otomatis maupun manual pada saat pengendalian robot. Dan robot yang ke dua diwajibkan untuk otomatis . Robot yang ke 2 biasa disebut dengan robot 'eco', dimana robot eco ini mempunyai tugas khusus yaitu harus mampu berjalan pada suatu track tertentu dengan hanya garis putih sebagai acuan posisi dari robot tersebut. Dengan syarat robot tersebut hanya mempunyai satu actuator saja yang digunakan untuk menentukan posisi dari robot tersebut.

Selama ini kebanyakan robot *line follower* menggunakan 2 aktuator untuk mengatur posisi dari robot tersebut.Namun pada perlombaan KRAI 2016 ini penggunaan 2 aktuator sebagai pengatur dari posisi robot tidak diperbolehkan.Sehingga para peserta KRAI harus memikirkansebuah desain robot yang hanya mempunyai 1 aktuator namun mampu mengatur posisi dari robot tersebut.Dan juga pengaturan posisi robot itu denganhanya menggunakan *feedback* dari sensor warna.Dimana sensor warna tersebut berfungsi untuk mendeteksi garis putih yang digunakan sebagai acuan dari pergerakan robot.Selama ini kebanyakan warna dari lapangan yang digunakan pada robot *line follower* adalah hitam dan putih saja. Namun pada Kontes Robot ABU Indonesia (KRAI) 2016 warna dari lapangan tidak hanya hitam dan putih saja melainkan terdapat juga warna seperti biru, orange, kuning, dan hijau Sehingga pembacaan data dari sensor warna akan lebih susah dari robot *line follower* biasa.

Salah satu solusi yang dapat digunakan untuk membuat sebuah robot yang hanya mempunyai 1 aktuator dan mampu mengatur posisi pergerakan dari robot tersebut adalah menggunakan motor DC servo sebagai *steering* dari robot. Pemilihan motor DC servo

sebagai *steering* robot pada Kontes Robot ABU Indonesia 2016 sangatlah tepat karena motor DC servo dapat mengatur sudut dari *steering* robot.Dan juga kontroler yang digunakan adalah kontroler PID, karena kontroler PID mempunyai respon yang cepat jika dibandingkan dengan kontroler lainnya. Posisi dari robot dapat kita atur kemiringan nya menggunakan motor DC servo. Didalam motor DC servo juga sudah dilengkapi dengan potensio meter sehingga ke presisian nya lebih bagus dari pada motor DC biasa. Solusi yang dapat digunakan untuk memudahkan pembacaan data dari warna lapangan adalah menggunakan sensor warna photodiode dan menggunakan LED yang RGB. Jadi ketika warna dari lapangan berbeda beda maka LED akan mengikuti dari warna lapangan tersebut sehingga data yang dihasilkan tidak terlalu dekat dan pengidentifikasian warna akan lebih mudah.

Sebuah metode yang simpel untuk mencari akar-akar persamaan karasteristik telah dikembangkan oleh W.R. Evans dan digunakan secara intensif dalam teknik kontrol. Metode ini disebut metode *root locus*, dimana salah satu akar dari persamaan karakteristik yang di plot untuk semua nilai dari parameter sistem [Ogata K, 1997]. Metode *root locus* digunakan untuk mencari parameter Kp, Ki dan Kd dengan cara melakukan perhitungan melalui matlab 2012.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka pembahasan skripsi ini ditekankan pada :

1. Bagaimana menemukan parameter PID menggunakan metode root locus ?
2. Bagaimana respon robot Eco ketika menggunakan kontroler PID?

1.3. Batasan Masalah

Berdasarkan masalah yang telah dirumuskan, maka perlu adanya batasan masalah yang berkaitan dengan perancangan dan penelitian. Beberapa batasan masalah nya adalah sebagai berikut :

1) Servo yang digunakan adalah servo hitech HS-5645MG
2) Mikrokontroler yang digunakan adalah Arduino Mega 2560
3) Robot menggunakan 4 roda.
4) Pembahasan ditekankan pada perancangan sebuah robot dan perancangan algoritma dari kontroler
5) Lintasan pengujian menyerupai bentuk lintasan yang digunakan pada lomba KRAI, namun lintasan pengujian datar dan hanya sampai lintasan river.

6) Robot yang akan diteliti hanya robot eco saja, tanpa robot hybrid.
7) Penelitian ditekankan hanya pada pengontrolan *steering* dari robot.
8) Motor DC flying yang digunakan hanya sebagai penggerak robot tanpa mengontrol posisi dari robot eco.
9) Penelitian tidak ditekankan pada pemodelan sistem.
10) Sensor garis yang digunakan adalah Photodiode dan LED yang digunakan adalah LED RGB(Red Green Blue).

1.4. Tujuan

Tujuan skripsi ini adalah untuk merancang sebuah robot Eco yang hanya menggunakan 1 buah actuator yaitu servo sebagai steering dengan *feedback* dari sensor warna dengan menggunakan kontroler PID untuk diterapkan pada Robot KRAI. Target yang harus dicapai adalah menghasilkan robot yang mampu mengatur posisi dari robot hanya dengan 1 aktuator dengan toleransi kesalahan pembacaan sensor warna sekecil mungkin dan memiliki respon yang baik serta stabil.

1.5. Manfaat Penelitian

Manfaat penelitian adalah sebagai referensi untuk tim robot ketika nantinya rule yang keluar dari ABU Robocon mempunyai kemiripan dengan penelitian ini.Selain itu penilitian ini juga dapat menjadi sumber referensi bagi yang ingin meneliti mengenai robot line follower.

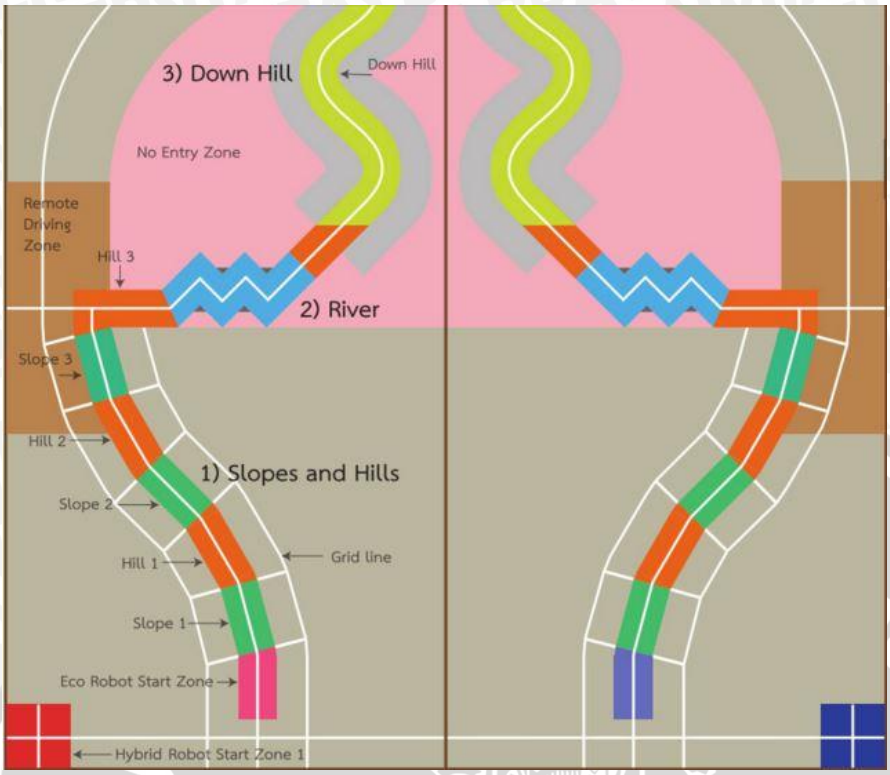
BAB II
TINJAUAN PUSTAKA

2.1. Kontes Robot ABU Indonesia (KRAI).

Kontes Robot ABU Indonesia (KRAI) adalah salah satu kategori yang dilombakan pada Kontes Robot Indonesia. Dimana rule yang digunakan pada Kontes Robot ABU Indonesia (KRAI) selalu mengacu pada ketentuan ABU Robocon. ABU(Asian Broadcasting Union) Robocon adalah lomba yang diadakan oleh persatuan penyiaran se Asia-Pasifik untuk mengembangkan minat dan bakat para mahasiswa dibidang robotika. Setiap tahun rule yang dikeluarkan oleh ABU Robocon selalu berbeda tergantung dari Negara yang akan menjadi tuan rumah. Contoh nya untuk tahun 2015 lombanya bertema "Robominton", dimana setiap mahasiswa diadu tingkat kreatifitas nya untuk membuat sebuah robot yang mampu bermain badminton.

Untuk tahun 2016 tema ABU robocon adalah "Clean Energy Recharging The World". Setiap tim diwajibkan membuat 2 buah robot yaitu robot eco dan robot hybrid. Robot hybrid bertugas untuk mendorong robot eco melewati sebuah lintasan yang telah diatur sedemikian rupa.Robot hybrid boleh dikontrol secara manual atau pun otomatis, sedangkan robot eco tidak diperbolehkan dikontrol secara manual namun harus dalam keadaan otomatis. Robot eco mempunyai misi membawa sebuah propeller melalui sebuah lintasan yang telah ditentukan sampai akhir lintasan. Lintasan yang digunakan dalam perlombaan dapat dilihat pada gambar 2.1 berikut.

Robot eco tidak mempunyai aktutor untuk bergerak sehingga membutuhkan bantuan dari robot hybrid untuk menggiringnya melewati lintasan yang telah ditentukan, pada robot eco hanya diperbolehkan menggunakan 1 aktuator untuk mengatur posisi dari dari robot eco.(ABU robocon, 2016). Tugas robot eco adalah membawa propeller untuk melintasi jalur track yang kecil dan hanya terdapat sebuah garis putih dimana sebagai referensi pergerakan dari robot eco tersebut, sehingga kinerja dari robot eco sangat bergantung dari sensor warna yang digunakan untuk mendeteksi lintasan. Pada Gambar 2.1 adalah bentuk lapangan yang digunakan pada ABU Robocon 2016.



Gambar 2.1 Lapangan perlombaan KRAI.

Sumber : ABU ROBOCON 2016 Rule (2016:15).

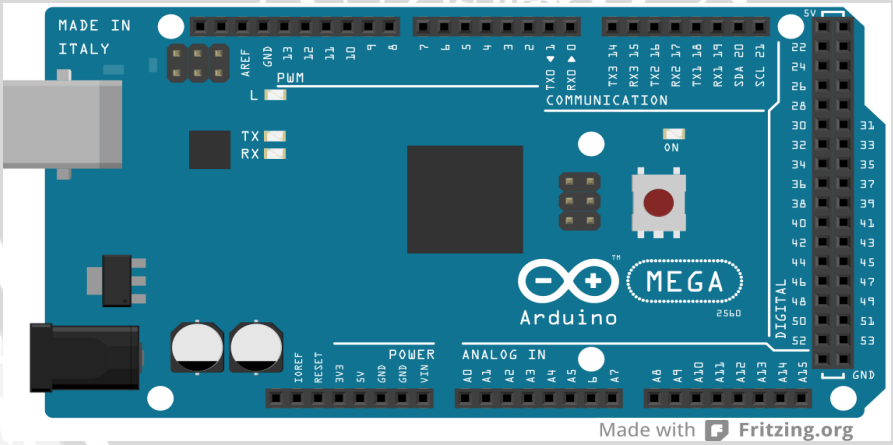
2.2. Arduino Mega 2560.

Arduino Mega 2560 adalah papan mikrokontroler berbasis ATmega 2560. Arduino mega 2560 mempunyai keunggulan dimana pin input/output nya berjumlah 54, dimana 15 pin dapat digunakan sebagai output PWM, 16 pin sebagai in/out analog, dan 4 pin sebagai UART(port serial hardware). Arduino mega mempunyai kecepatan clock sebesar 16MHz, koneksi USB, jack power, header ICSP, dan tombol reset. Semua fitur ini diperlukan untuk mendukung mikrokontroler berkomunikasi atau tsistem transfer data dari computer atau peralatan elektronik lainnya . Cukup dengan menghubungkannya ke computer melalui kabel USB atau power dihubungkan dengan adaptor AC-DC atau baterai untuk mulai mengaktifkannya. Pada Tabel 2.1 dapat dilihat spesifikasi Hardware Arduino Mega 2560.

Tabel 2.1 Spesifikasi Hardware Arduino Mega 2560.

Keterangan	Spesifikasi Hardware
Mikrokontroller	ATmega 2560
Operating Voltage	5V
Input Voltage(recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	54 (15 pin dapat digunakan sebagai output PWM)
Analog Inputs Pin	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Arduino mega sudah dilengkapi dengan bootloader yang mempermudah kita untuk mengupload sebuah program pada arduino mega. Arduino mega 2560 juga dilengkapi resettable polyfuse yang mampu melindungi USB dari komputer dari short circuit dan overcurrent.jika arus lebih besar dari 500 mA melalui port USB maka fuse akan secara otomatis memutus koneksi sampai overload atau short circuit hilang. Tampak atas dari arduino mega 2560 dapat dilihat pada Gambar 2.2



Gambar 2.2 Gambar Arduino mega 2560.

Sumber: https://www.arduino.cc/en/Main/ArduinoBoardMega2560.

2.3. Photodiode.

Photodiode adalah salah satu jenis diode, dimana nilai dari resistansinya dapat berubah-ubah.Prinsip kerja dari photodiode adalah mendeteksi cahaya sekitar dan merubahnya menjadi sebuah resistansi.Perubahan resistansi disebabkan oleh cahaya yang ditangkap oleh photodiode. Ketika cahaya yang ditangkap oleh photodiode

semakin gelap maka nilai resistansinya akan semakin besar, namun ketika cahaya yang ditangkap oleh photodiode semakin terang akan semakin kecil nilai resistansinya. Resistansi akan berpengaruh pada arus yang lewat, semakin besar resistansi maka arus nya akan semakin kecil. Photodiode terbuat dari bahan semikondukto.Biasanya menggunakan silicon (Si), gallium arsenide (GaAs), dll.

Pada photodiode terdapat dua Kaki yang menunjukkan polaritas dari LED tersebut. Kaki yang lebih panjang menunjukkan polaritas (+) atau biasa disebut polaritas anoda, sementara untuk kaki yang lebih pendek adalah polaritas katoda atau (-).Photodiode mempunyai respon yang sangat cepat terhadap cahaya (W.Bolton:1999). Bentuk dari Photodiode dapat dilihat pada Gambar 2.3

Gambar 2.3 Photodiode.

2.4. LED RGB (Red Green Blue)

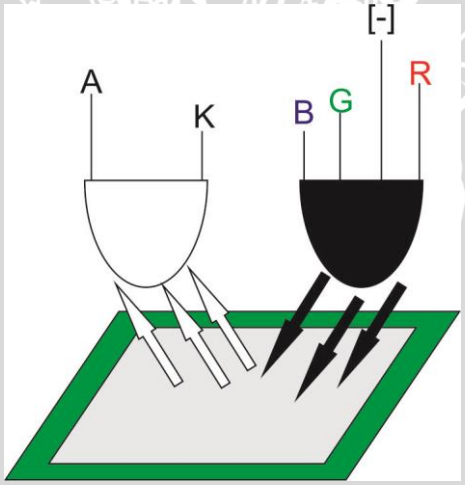
LED (Light Emitting diode) adalah komponen elektronik yang mampu memancarkan cahaya.LED termasuk dalam salah satu keluarga diode namun terbuat dari bahan semikonduktor. Cara kerja LED hampir sama dengan diode dimana mempunyai 2 kutub yaitu kutub Positif (P) dan kutub negative (N). LED akan memancarkan cahaya apabila dialiri tegangan maju(bias forward) dari anoda ke katoda.

LED RGB(Red Green Blue) adalah LED yang dapat memancarkan 3 warna dasar yaitu merah, hijau, dan biru. Dari kombinasi 3 warna tersebut dapat berubah menjadi warna lain, sehingga tidak membutuhkan terlalu banyak LED untuk memberi warna yang berbeda beda sesuai dengan yang kita inginkan. LED RGB mempunyai 4 pin yang terdiri dari 3 pin untuk mengontrol warna R-G-B dan 1 pin untuk common katode. Pemilihan LED RGB sangat berguna bagi penelitan dikarenakan warna dasar lintasan yang berbeda-beda maka kita bisa memilih warna yang tepat untuk digunakan pada percobaan nantinya. Bentuk dari LED RGB dapat dilihat pada Gambar 2.4

Gambar 2.4 LED RGB.

2.5. Sistem Kerja sensor garis.

LED RGB digunakan untuk pengirim cahaya ke garis untuk dipantulkan, setelah itu dibaca oleh sensor photodiode. Dimana nantinya pemilihan warna LED RGB akan berpengaruh terhadap pembacaan sensor photodiode. Sehingga sensor photodiode yang digunakan harus sejajar dengan LED RGB yang digunakan. Untuk ilustrasi mekanisme sensor garis dapat dilihat pada Gambar 2.5



Gambar 2.5 Prinsip Kerja Sensor Garis

Sensor photodioda berfungsi untuk mendeteksi garis putih yang berada pada lintasaan pengujian. Sehingga nantinya robot mampu mengetahui waktu yang tepat untuk berbelok ke kanan maupun ke kiri.

2.6. Kontroler

Pada sebuah plant kontroler adalah sebuah system yang sangat dibutuhkan untuk mencapai suatu setpoint yang kita inginkan.Jika di ibaratkan sebagai anggota tubuh kontroler adalah otak kita dimana pusat kendali dari seluruh tubuh kita berada di otak.Dengan memanfaatkan indera kita sebagai sensor sehingga kita dapat memberikan respon yang baik untuk tubuh kita.Sama hal nya dengan kontroler yang memberikan respon apabila terdapat eror pada system kita.

Sistem kontrol berumpan balik adalah sistem kontrol yang cenderung menjaga hubungan yang telah ditentukan antara keluaran dan masukan acuan dengan membandingkanya dan menggunakan selisihnya sebagai alat pengontrolan. (Leksono, E:1995)

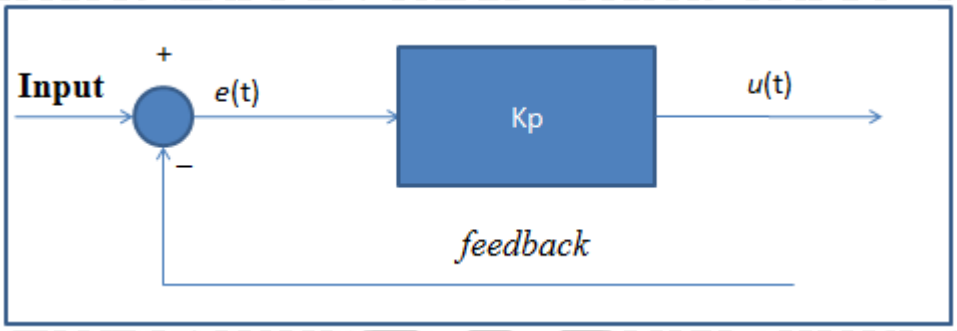
Prinsip kerja kontroler adalah membandingkan nilai output plant dengan nilai setpoint, menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan kesalahan (Ogata K, 1997).

2.6.1. Kontroler Proportional

Kontroler Proposional mempunyai karakteristik menurunkan nilai rise time dari system dan juga menurunkan error steady state namun tidak menghilangkan nya. Kontroler ini mempunyai kelemahan yaitu meningkatkan overshoot sehingga dibutuhkan kontroler lainnya untuk menutupi kekurangan nya. Dapat dikatakan juga bahwa keluaran kontroler proportional merupakan perkalian antara konstanta proporsional dengan masukannya. Kontroler proporsional mempunyai ciri-ciri sebagai berikut :

a. Ketika nilai Kp kecil, maka kontroler hanya akan melakukan koreksi kesalahan yang kecil dan mengakibatkan respon system lambat.
b. Ketika nilai Kp dinaikkan, respon system menunjukkan semakin cepat mencapai keadaan mantabnya.

Kontroler proportional, keluarannya selalu sebanding dengan masukannya. Kontroler proportional selalu memerlukan *error* untuk menghasilkan *output*.



Gambar 2.6 Blok Diagram Kontroler Proportional.

Sumber: Ogata (1997)

Keterangan :

Kp = *gain* proporsional

e(t) = sinyal *error*

u(t) = *output* kontroler

hubungan antara output kontroler u(t) dan sinyal error e(t) ditunjukkan dalam persamaan berikut :

$$u(t) = Kp\ e(t) \dots\dots (2-1)$$

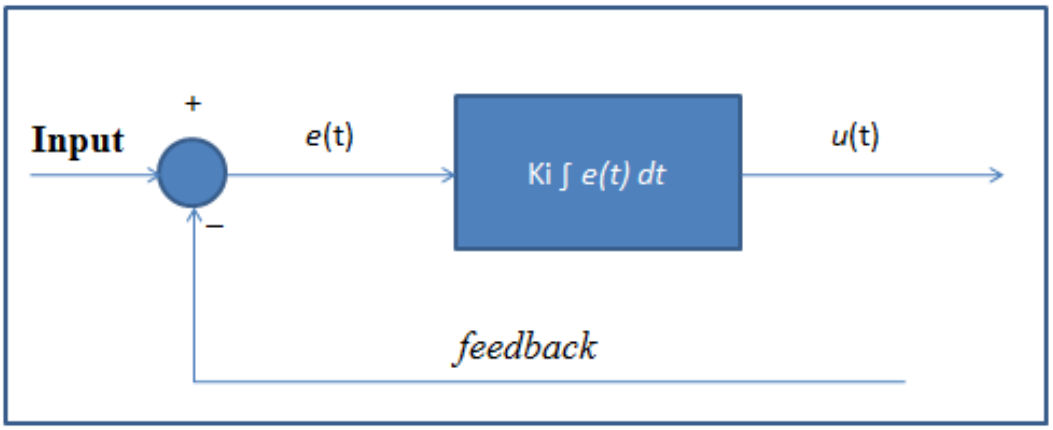
Jika di transformasikan laplace persamaan akan menjadi :

$$U(s) = Kp.E(s) \dots\dots (2-2)$$

2.6.2. Kontroler Integral

Kontroler integral memiliki sifat seperti sebuah operasi integral, output kontroler sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal *error*.

Kontroler integral mempunyai karakteristikmenurunkan rise time system, meningkatkan overshoot dan juga menaikkan settling time system, sehingga dapat mengakibatkan respon yang tidak stabil .Kontroler integral berbeda dengan kontroler lainnya dikarenakan berfungsi sebagai penghilang sinyal *error* dalam *steady state*.output kontroler merupakan penjumlahan dari perubahan sinyal *error*. Konstanta integral Ki yang besar akan mempercepat hilangnya *offset*. Tetapi semakin besar nilai konstanta Ki akan mengakibatkan peningkatan osilasi dari sinyal keluaran kontroler. Berikut ini adalah gambar diagram blok kontroler integral :



Gambar 2.7 Blok Diagram Kontroler Integral.

Sumber: Ogata (1997)

Keterangan :

Ki = gain *integral*

e(t) = sinyal *error*

u(t) = *output* kontroler

Persamaan kontroler Integral (I) dapat ditunjukkan dalam persamaan berikut (OgataK,1997).

$$\frac{du(t)}{dt} = Ki\, e(t) \dots\dots (2-3)$$

$$u(t) = Ki \int_0^t e(t) \dots\dots (2-4)$$

Jika di transformasikan laplace persamaan akan menjadi :

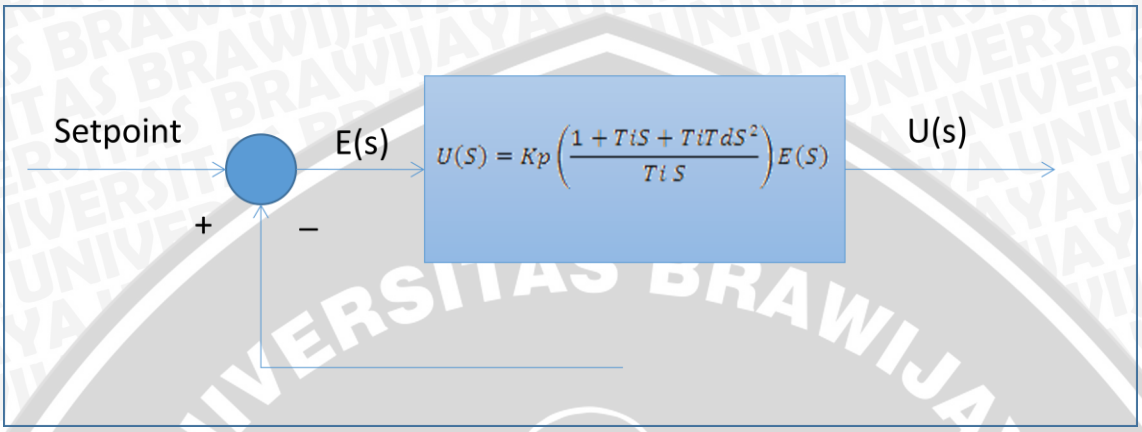
$$U(s) = \frac{\text{Ki}}{\text{s}}.E(s) \dots\dots (2-5)$$

2.6.3. Kontroler Derivative

Kontroler *derivative*memiliki sifat seperti operasi *derivative* pada matematika. Perubahan yang mendadak pada masukan kontroler , akan mengakibatkan perubahan yang sangat besar dan cepat. Kontroler derivative tidak dapat menghasilkan keluaran apabila tidak ada perubahan pada masukannya (berupa sinyal *error*). Kontroler ini tidak akan menghasilkan *output* saat sinyal *error* konstan sehingga tidak akan mempengaruhi keadaan mantap. Kontroler *derivative*biasanya dipakai untuk mempercepat respon awal suatu system, tetapi tidak memperkecil kesalahan pada keadaan tunak.Selain itu kontroler tidak dapat digunakan apabila tidak ada kontroler lainnya dikarenakan kontroler ini hanya efektif ketika periode transient saja [Ogata. K:217].

2.6.4. Kontroler PID.

Kombinsasi dari kontroler proportional, kontroler integral, dan kontroler derivative disebut kontroler PID(Proportional Integral Derivative). Kombinasi dari ketiga kontroler mempunyai manfaat dari masing-masing kontroler [Ogata, K:218]. Blok diagram kontroler PID dapat dilihat pada Gambar 2.8



Gambar 2.8 Blok diagram kontroler PID

Sumber:Ogata(1997)

Keterangan :

Kp = *gain* Proportional

Ki = *gain* Integral

Kd = *gain*Derivative

e(t) = sinyal *error*

u(t) = *output* kontroler

Persamaan umum kontroler PID dapat dilihat pada persamaan berikut [Ogata K, 1997]:

$$u(t) = Kp\,e(t) + \frac{Kp}{Ti}\int_0^t e(t)dt + Kp\,Td\,\frac{de(t)}{dt} \dots\dots (2-6)$$

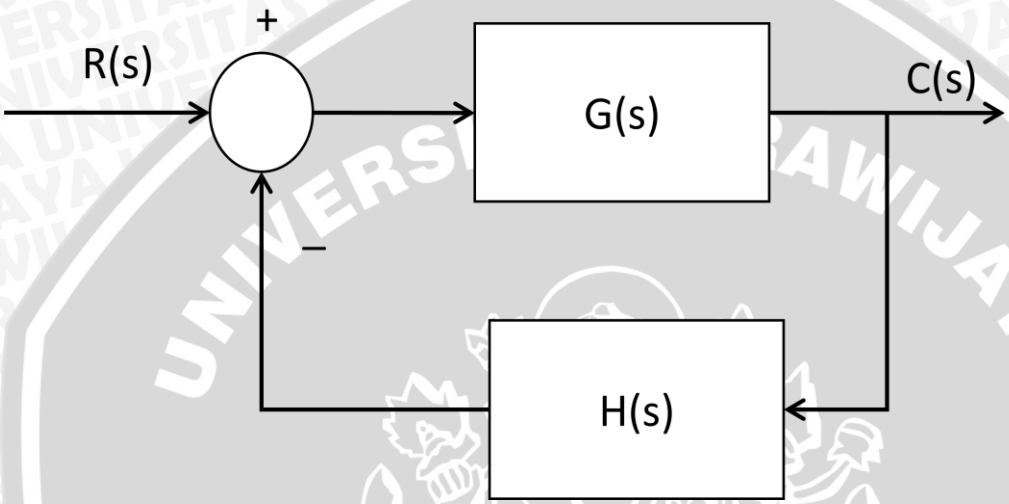
Jika di transformasikan laplace persamaan akan menjadi :

$$U(s) = Kp(1 + \frac{1}{Ti\,S} + Tds)E(s) \dots\dots (2-7)$$

$$U(S) = Kp\left(\frac{1 + TiS + TiTdS^2}{Ti\,S}\right)E(S) \dots\dots (2-8)$$

2.7. Metode Root Locus.

Sebuah metode yang simpel untuk mencari akar-akar persamaan karasteristik telah dikembangkan oleh W.R. Evans dan digunakan secara intensif dalam teknik kontrol. Metode ini disebut metode *root locus*, dimana salah satu akar dari persamaan karakteristik yang di plot untuk semua nilai dari parameter sistem [Ogata K, 1997]. Blok Diagram loop tertutup dapat kita lihat pada gambar 2.9.



Gambar 2.9 Sistem Kontrol Loop tertutup

Sumber : Ogata, K. (1997)

Fungsi alih pada gambar 2.9 dapat dilihat pada persamaan berikut:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \dots\dots (2-9)$$

Persamaan karakteristik untuk sistem kontrol loop tertutup dapat ditemukan dengan menyamakan denominator persamaan tersebut sama dengan nol.[Ogata, K:1997] berikut adalah persamaan karakteristik dari fungsi alih tersebut :

$$1 + G(s)H(s) = 0$$

Atau

$$G(s)H(s) = -1 \dots\dots (2-10)$$

Berikut adalah persamaan yang digunakan untuk mencari angle dan magnitude.

Kondisi sudut:

$$\angle G(s)H(s) = \pm 180°(2k+1)(\mathrm{k} = 0,1,2,\dots\dots) \dots\dots (2-11)$$

Kondisi Magnitude:

$$|G(s)H(s)| = 1 \dots\dots (2-12)$$

Fungsi alih kontroler PID setelah di transformasi laplace dapat dilihat pada persamaan 2-13 berikut :

$$G(s) = Kp + \frac{Ki}{s} + Kd\ s \dots\dots (2-13)$$

Perhitungan persamaan 2-13 ketika akan di substitusikan dengan pole s1 dapat dilihat pada persamaan berikut :

$$G(s_1) = \frac{1}{H(s_1)} e^{j(\Pi - \Psi)} \dots\dots (2-14)$$

Selanjutnya substitusi persamaan 2-14 ke persamaan 2-13 didapatkan:

$$\frac{e^{j(\Pi-\Psi)}}{H(s_1)} = Kp + \frac{Ki}{s} + Kd\ s \dots\dots (2-15)$$

Substitusikan s1 dengan persamaan berikut :

$$s_1 = |s_1| e^{j\beta} \dots\dots (2-16)$$

Hasil substitusi dapat dilihat pada persamaan 2-17:

$$Kd|s_1|^2(cos2\beta + jsin2\beta) + Kp|s_1|(cos\beta + jsin\beta) + Ki$$
$$= \frac{|s_1|}{|H(s_1)|}[\cos(\beta + \Pi - \Psi) + jsin(\beta + \Pi - \Psi)] \dots\dots (2-17)$$

Meyamakan real dan imajiner dengan imajiner, didapatkan hasil seperti pada persamaan 2-18 :

$$\begin{bmatrix} |s_1|^2 & |s_1|cos\beta \\ |s_1|^2 & |s_1|sin\beta \end{bmatrix} \begin{bmatrix} Kd \\ Kp \end{bmatrix} = \begin{bmatrix} \frac{|s_1|}{H(s_1)}\cos(\beta + \Pi + \Psi) - Ki \\ \frac{|s_1|}{H(s_1)} sin\ (\beta + \Pi + \Psi) \end{bmatrix} \dots\dots (2-18)$$

Atau dapat ditunjukkan pada persamaan 2-19 :

$$\begin{bmatrix} |s_1|^2 & |s_1|cos\beta \\ |s_1|^2 & |s_1|sin\beta \end{bmatrix} \begin{bmatrix} Kd \\ Kp \end{bmatrix} = \begin{bmatrix} -\frac{|s_1|}{H(s_1)}\cos(\Psi - \beta) - Ki \\ \frac{|s_1|}{H(s_1)} sin\ (\Psi - \beta) \end{bmatrix} \dots\dots (2-19)$$

Pada persamaan tersebut dapat dilihat bahwa untuk menentukan parameter Kp, Ki, dan Kd harus ditentukan terlebih dahulu salah satu parameter nya. Sedangakan pada kontroler PI atau PD parameter yang tidak di masukkan disamakan dengan nilai nol.

2.8. Diskritisasi

Terdapat beberapa cara yang dapat digunakan untuk proses diskritisasi. Beberapa cara tersebut adalah *backward difference*, *forward difference*(metode euler) dan *bilinear transformation*. Beberapa metode tersebut hanya merupakan pendekatan, sehingga hasilnya tidak akan sama persis dengan bentuk analog. Hal ini dikarenakan bentuk diskrit selalu ada sebagian informasi yang hilang, yaitu informasi yang terjadi antara satu cuplikan dengan cuplikan lainnya, walaupun frekuensi cuplik yang digunakan tinggi (Houpis,C.1992).

Berikut langkah-langkah yang dilakukan untuk melakukan diskritisasi :

1. Mencari algoritma analog dalam bentuk transformasi laplace.
2. Melakukan diskritisasi dengan menggunakan salah satu metode untuk merubah bentuk kedalam transformasi Z, berikut adalah beberapa metode diskritisasi yang bisa digunakan :

- Backward difference :

$$S = \frac{1 - z^{-1}}{Ts} \dots\dots (2-9)$$

- Forward Difference:

$$S = \frac{1 - z^{-1}}{Ts\, z^{-1}} \dots\dots (2-10)$$

- Bilinear Transform:

$$s = \frac{2(1 - z^{-1})}{Ts\,(1 + z^{-1})} \dots\dots (2-11)$$

Dengan :

Ts= Periode cuplik

Setelah didapatkan algoritma persamaan dalam bentuk z, maka selanjutnya mengubah persamaan z dalam bentuk persamaan time domain(persamaan beda), yaitu dengan mengubah operator z^n menjadi n kali waktu delay (z^{-1}berarti 1 kali waktu delay, z^{-2} berarti 2 kali waktu delay dan seterusnya).

2.9. Motor DC Servo HS-5685 MH

Servomekanisme(servomechanism) adalah sistem kontrol berumpan balik dengan keluaran berupa posisi, kecepatan, atau percepatan mekanik. Oleh karena, istilah servomekanisme dan sistem pengontrolan posisi(atau kecepatan atau percepatan) adalah sinonim. Servomekanisme banyak digunakan dalam industri modern. Sebagai contoh,

operasi mesin perkakas yang seluruhnya automatik, bersama-sama dengan instruksi yang telah diprogram terlebih dahulu, dapt dibuat dengan menggunakan servomekanisme (Leksono, E:1995).

Motor DC servo adalah salah satu jenis motor DC yang dapat diputar dalam besaran sudut tertentu sesuai dengan apa yang kita inginkan. Servo terdiri dari 4 komponen penting didalam nya yaitu motor DC, gear, feedback device(potensiometer), dan rangkaian pengontrol. Motor akan memutar poros servo melalui beberapa gear yang sudah di kopel, dan juga yang telah dikopel dengan potensiometer. Tugas dari potensiometer adalah mengirim sinyal kepada rangkaian pengontrol untuk mengetahui sudut pada servo.

Servo dapat digerakkan dengan mengirimkan pulsa tegangan 5V DC yang diulang setiap 20 milisecond.Panjang pulsa menentukan posisi sudut dari motor DC servo.Keunggulan dari motor DC servo adalah pada saat kecepatan tinggi tidak menimbulkan bunyi yang bising, dengan ukuran yang relative tinggi mempunyai torsi yang cukup tinggi, dan penggunaan arus listrik sebanding dengan beban yang diberikan. Servo HS-5685 MH telah didesain dengan catu 2-cell 7,4V Lipo batterai untuk memberikan performa yang optimal. Spesifikasi dari servo HS-5685 MH dapat dilihat pada Tabel 2.2

Tabel 2.2Spesifikasi servo HS-5685 MH

No.	Spesifikasi	Spesifikasi detail
1.	Control system	+pulse Width Control 1500usec
2.	Operating Voltage Range	4.8 V to 7.4 V
3.	Operating Temperature Range	-20° to +60° C (-4°F to +140°F)
4.	Operating speed (6 V)	0.2 sec/60° at no load
5.	Operating speed (7,4 V)	0.17 sec/60° at no load
6.	Stall torque (6 V)	157 oz/in (8,8 kg.cm)
7.	Stall torque (7,4 V)	179 oz/in (12,9kg.cm)
8.	Gear type	Metal Gear
9.	Direction	Clockwise/Pulse Traveling 1500 to 1900usec
10.	Potentiometer Drive	6 slider Indirect Drive
11.	Weight	2.1oz (60g)
12.	Required Pulse	3-5 Volt peak to peak square wave

Gambar 2.10 Servo HS-5685 MH.

BAB III
METODOLOGI PENELITIAN

Untuk menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian maka diperlukan langkah-langkah metode untuk menyelesaikan masalah tersebut.Adapun langkah-langkah yang perlu dilakukan untuk merealisasikan sistem yang dirancang adalah penentuan studi literatur, spesifikasi alat, perancangan dan pembuatan alat, pengujian alat, dan pengambilan kesimpulan.

3.1. Studi Literatur

Studi literatur dilakukan dengan mencari kajian pustaka dari berbagai sumber yang sesuai dengan judul penelitian. Kajian tersebut berkaitan dengan metode*root locus*, arduino mega 2560, PID kontroler, Robot *line follower*, dan bluetooth CZ-HC-05.

3.2. Spesifikasi Alat

Spesifikasi alat secara global ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang direncanakan adalah sebagai berikut :

1. Servo yang digunakan adalah Hitech HS-5685 MH
2. Mikrokontroler yang dugunakan adalah Arduino Mega 2560
3. Jumlah sensor garis adalah 6 buah.
4. Roda yang digunakan adalah 4
5. Tipe Bluetooth yang digunakan adalah CZ- HC- 05
6. Set point yang di inginkan adalah 95°.
7. Bluetooth hanya digunakan untuk pembacaan data serial saja.
8. Driver Motor yang digunakan adalah L298N yang digunakan untuk mengatur PWM agar gerak dari robot hanya berjalan lurus tanpa mengkontrol posisi dari robot.
9. Terdapat 2 sumber catu daya yang digunakan yaitu sumber baterai lippo 12V dan baterai lippo 7,4V. Catu daya 12 V digunakan untuk motor DC yang digunakan untuk mendorong robot untuk bergerak lurus, serta digunakan sebagai catu arduino dan sensor dimana dilewatkan ke step down terlebih dahulu agar tegangan menjadi 5V. Catu daya 7,4 V digunakan untuk catu daya motor DC servo.

3.3. Spesifikasi Desain

Psesifikasi desain yang di inginkan pada posisi untuk robot eco mempunyai sepsifikasi sebagai berikut :

1. Motor DC servo yang digunakan adalah servo HS-5685 dikarenakan servo ini bekerja optimal pada tegangan 7,4 V.
2. Motor DC yg digunakan untuk mendorong robot adalah motor DC flying dikarenakan mempunyai torsi sebesar 4,8 kg dan optimal pada catu daya 12V.
3. Duty cycle maksimal Motor DC flying yang digunakan adalah 91%, dikarenakan pada kecepatan tersebut robot mampu bekerja optimal.
4. Sudut error maksimal yang di inginkan adalah 0°. Namun ketika pengujian maksimal sudut yang dihasilkan adalah 40°.
5. Sensor warna yang digunakan adalah photodiode. Dikarenakan sensor tersebut sangat sensitif pada cahaya.
6. LED yang digunakan adalah LED RGB. Dikarenakan LED ini mempunyai kombinasi warna yang lebih banyak dari LED satu warna. Terdapat 7 kombinasi warna yang bisa digunakan untuk memilih warna yang tepat.

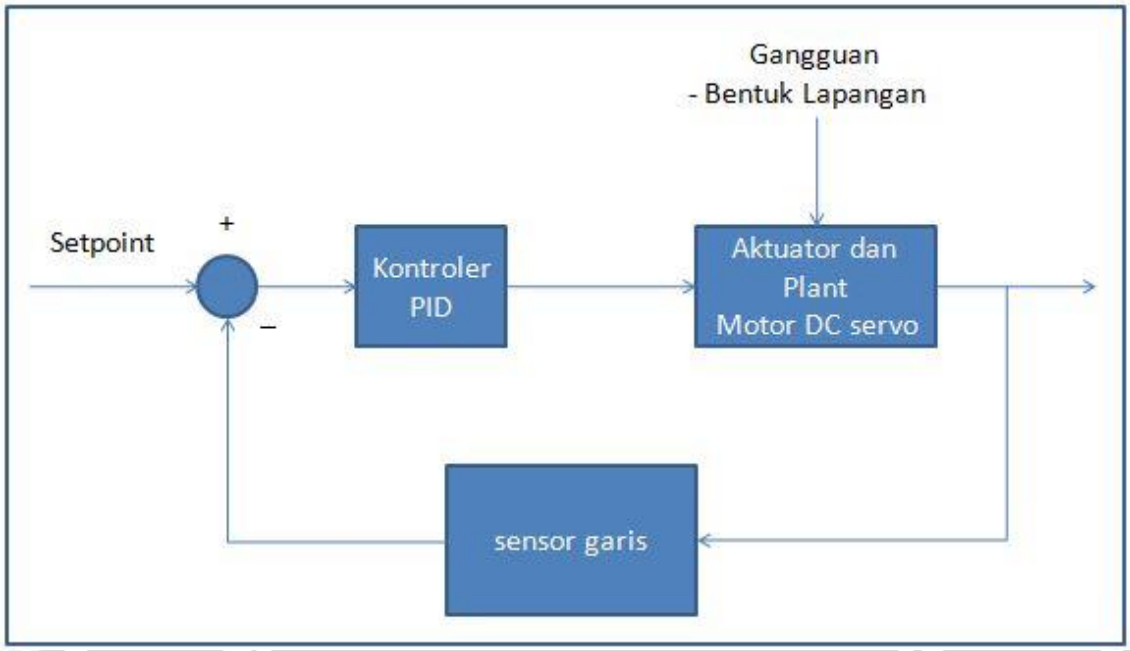
3.4. Perancangan dan Pembuatan Alat

Perancangan dilakukan untuk mempermudah pada saat proses pembuatan alat nantinya. Perancangan dan pembuatan alat dilakukan dengan beberapa tahap yaitu :

1. Perancangan Blok diagram
2. Konfigurasi Pin Mikrokontroller Arduino Mega 2560.
3. Perancangan Perangkat Keras
4. Sistem Kerja Elektrik Alat
5. Perancangan Software.
6. Diskritisasi.

3.4.2. Perancangan Diagram Blok

Pada perencanaan awal sebelum penelitian diperlukan blok diagram yang mampu menjelaskan system secara keseluruhan. Dan nantinya diharapkan dari blok diagram ini dapat mempermudah dalam penelitian. Blok diagram dapat dilihat pada Gambar 3.1



Gambar 3.1 Blok Diagram Sistem

Pada blok diagram yang ditunjukkan Gambar 3.1 tersebut set pointnya adalah sudut yang digunakan agar robot menjadi lurus dalam hal ini adalah 95°. Dimana kontroler yang digunakan adalah kontroler PID. Sementara aktuator yang digunakan adalah servo HS-5685 MH, untuk plant yang dikontrol adalah posisi dari robot.

Gangguan (disturbances) adalah suatu sinyal yang cenderung mempunyai pengaruh yang merugikan pada harga keluaran sistem. Jika suatu gangguan dibangkitkan didalam sistem maka disebut *internal* sedangkan gangguan gangguan eksternal dibangkitkan diluar sistem dan merupakan suatu masukan (Laksono,E:1995) gangguan pada plant ini bisa berupa warna lapangan dan juga bentuk dari lapangan yang berbeda-beda. Semakin besar sudut perubahan pada lintasan maka semakin besar gangguan yang diberikan terhadap plant. Sementara sensor yang digunakan adalah sensor photodioda yang digunakan untuk mendeteksi garis putih yang berada dalam lintasan pegujian sistem.

3.4.3. Konfigurasi Pin Mikrokontroller Arduino Mega 2560

Pada Tabel 3.1 berikut ini adalah pin yang digunakan pada arduino mega :

Tabel 3.1 Konfigurasi Pin Arduino mega 2560

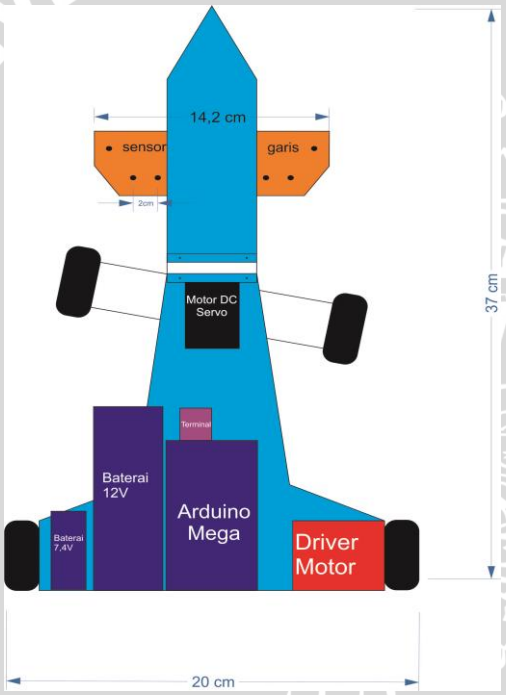
No.	Pin	Fungsi
1	A1	Data sensor garis 1
2	A2	Data sensor garis 2
3	A3	Data sensor garis 3
4	A4	Data sensor garis 4
5	A5	Data sensor garis 5
6	A6	Data sensor garis 6
7	D2	Data servo

No.	Pin	Fungsi
8	A9	Data Red
9	A11	Data Green
10	A13	Data Blue

3.4.4. Perancangan Perangkat Keras

Perancangan perangkat keras disini adalah mendesain bentuk robot dan memposisikan peletakan elektrik yang digunakan serta sensor yang digunakan agar penelitian nantinya dapat lebih mudah dilakukan. Desain dan ukuran dari robot dapat dilihat pada Gambar 3.2.

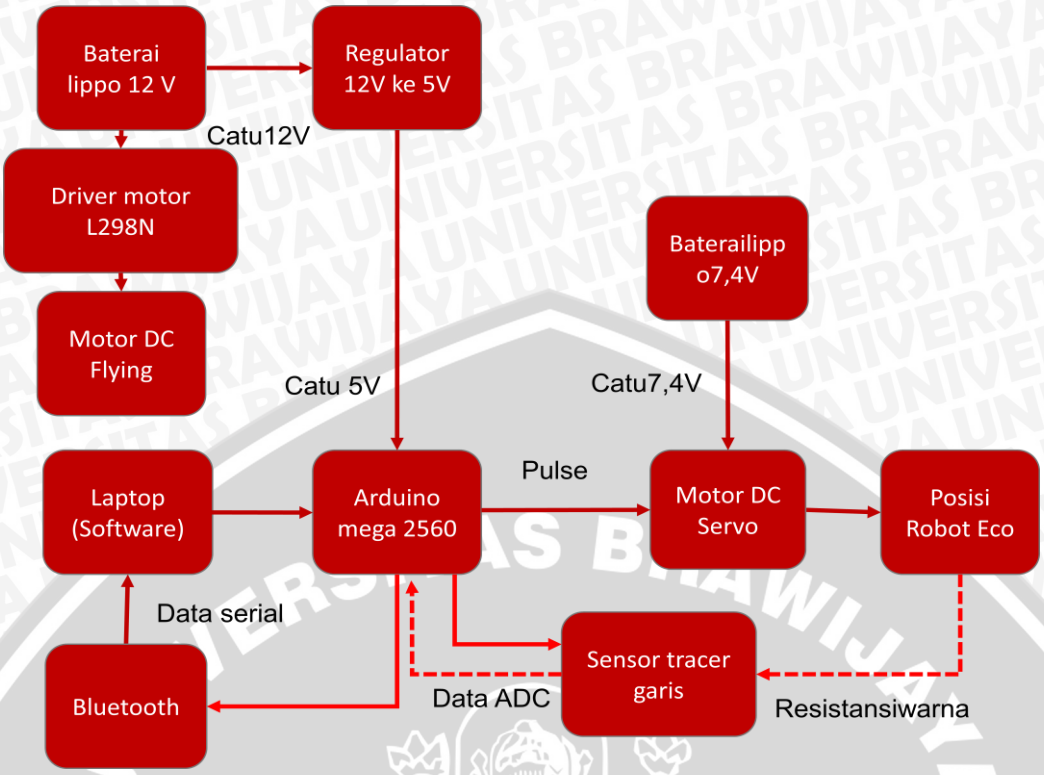
Pada Gambar 3.2 sensor garis yang digunakan untuk robot tidak menempel pada roda depan robot yang dikopel langsung dengan motor DC servo. Dimana sensor garis yang digunakan tetap dan tidak bergerak namun menempel pada body chasis utama robot. Pada Gambar 3.2 berikut ini adalah desain robot eco :



Gambar 3.2 Desain Robot

3.4.5. Sistem Kerja Elektrik Alat

Sistem kerja alat ini dibuat agar nantinya mempermudah kita dalam pembuatan robot eco. Pada Gambar 3.3 berikut ini adalah sistem kerja elektrik pada robot mulai dari catu daya hingga pengiriman data serial ke laptop.



Gambar 3.3 Sistem Kerja Elektrik Alat

Pada Gambar 3.3 robot eco yang digunakan dalam pembuatan ini mempunyai 2 catu daya yaitu baterai lippo 12V dan lippo 7,4V. Catu 12V digunakan untuk mencatu motor DC yang digunakan sebagai motor pendorong agar robot mampu berjalan lurus kedepan, selain itu catu 12V yang sudah di step down menjadi 5v digunakan sebagai catu arduino dan juga rangkaian sensor garis yang digunakan. Pada gambar juga dijelaskan bahwa pengiriman data dari robot yang digunakan menggunakan bluetooth.

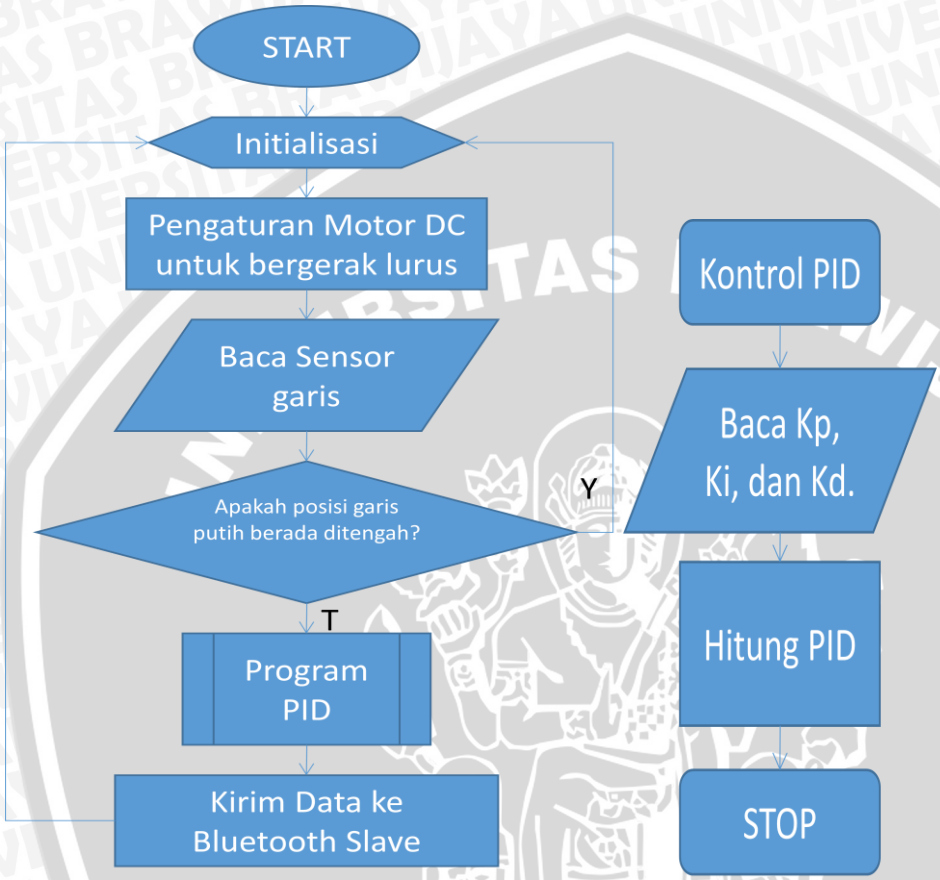
3.4.6. Perancangan Software

Perancangan software dilakukan agar nanti nya robot mampu bergerak sesuai dengan apa yang kita inginkan. Perancangan software dilakukan dalam beberapa tahap sehingga untuk mempermudah dalam proses pengerjaan. Berikut adalah beberapa tahap yang dilakukan :

1. Perancangan Flowchart sistem
2. Perancangan kontroler
3. Penentuan fungsi alih motor DC servo
4. Penentuan parameter kontroler PID menggunakan metode *root locus*
5. Diskritisasi.

3.4.6.1 Flowchart sistem

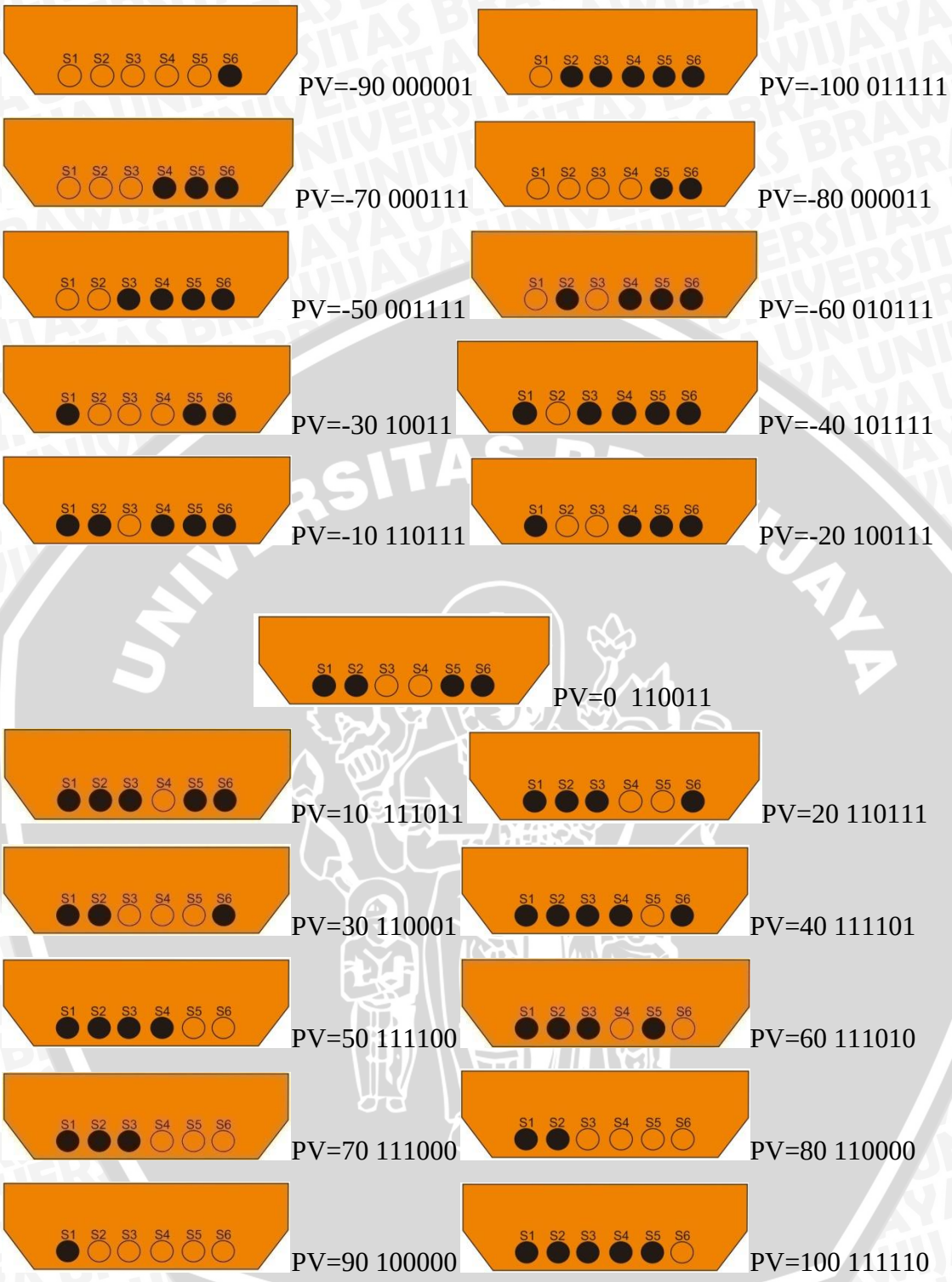
Flowchart adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan dari proses dan alur pemikiran dari pembuat sistem. Pada Gambar 3.3 berikut ini adalah flowchart sistem secara keseluruhan :



Gambar 3.4 Flowchart sistem secara keseluruhan

3.4.6.2 Perancangan Kontroler

Perancangan kontroler disini bertujuan untuk membuat algoritmayang nantinya akan digunakan pada robot pada saat pengujian sistem secara keseluruhan. Perancangan kontroler disini dilakukan dengan cara memberikan nilai PV (position value) kepada masing-masing kombinasi yang dihasilkan oleh sensor garis. Pada penelitian ini didapatkan 10 kombinasi yang terjadi selama percobaan Pada Gambar 3.5 berikut ini adalah kombinasi yang dihasilkan oleh sensor garis :



Gambar 3.5 Position Value(PV) pada sensor garis.

Dari 10 kombinasi diatas setpoint berada pada PV 0 yaitu dengan kombinasi sensor garis 110011. Barulah dari nilai PV tersebut nantinya akan masuk kedalam rumus PID dan dilakukan penghitungan. Pemberian nilai PV disini dilakukan dengan cara trial dan

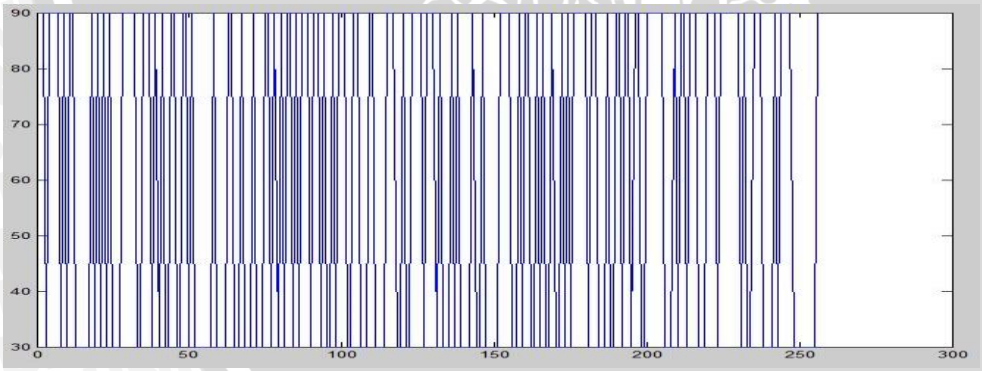
error dimana semakin besar PV yang diberikan maka semakin besar pula sudut yang digunakan pada robot.

Pada gambar 3.5ketika sensor mendeteksi garis putih dapat diindikasikan dengan nilai 0, namun ketika sensor mendeteksi warna selain warna putih akan di indikasikan dengan nilai 1 atau pada gamabr diatas berwarna hitam.

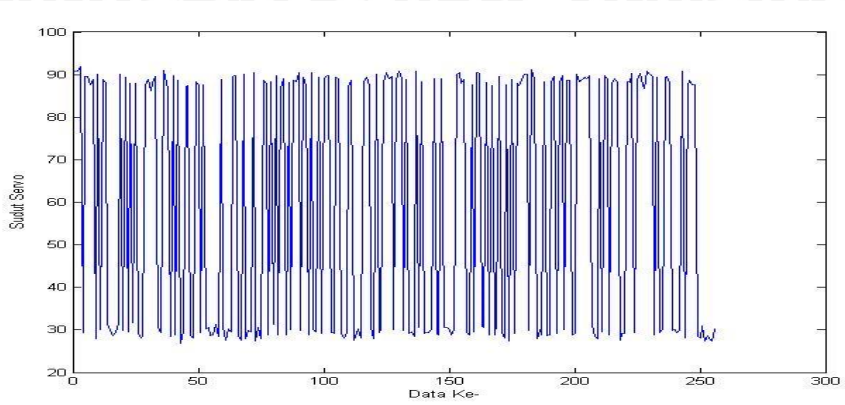
3.4.6.3 Penentuan Fungsi Alih Motor DC Servo

Untuk mampu mengkontrol robot dengan baik sehingga mempunyai performa yang bagus maka perlu dicari fungsi alih dari sistem itu sendiri, yang nantinya ketika sudah didapatkan fungsi alih dari motor DC servo barulah dapat dicari parameter Kp, Ki, dan Kd. Sehingga nantinya akan digunakan pada program utama robot. penentuan fungsi alih dari motor dc servo dapat dilakukan dengan cara pemodelan dimana untuk memodelkan nya kita harus membangkitkan sinyal *Pseudo Random Binary Sequence* (PRBS). Beberapa langkah yang dilakukan untuk membangkitkan sinyal PRBS adalah sebagai berikut :

1. Mencari nilai yang linear pada Pulsa High (Ton) motor DC servo.
2. Memasukkan batas atas dan batas bawah dari nilai yang linear tersebut.
3. Sinyal PRBS yang telah dibangkitkan dan datanya telah didapatkan maka nantinya akan menjadi input dan output pada identifikasi sistem. Gambar input dapat dilihat pada Gambar 3.6dan Gambar output PRBS dapat dilihat pada Gambar 3.7

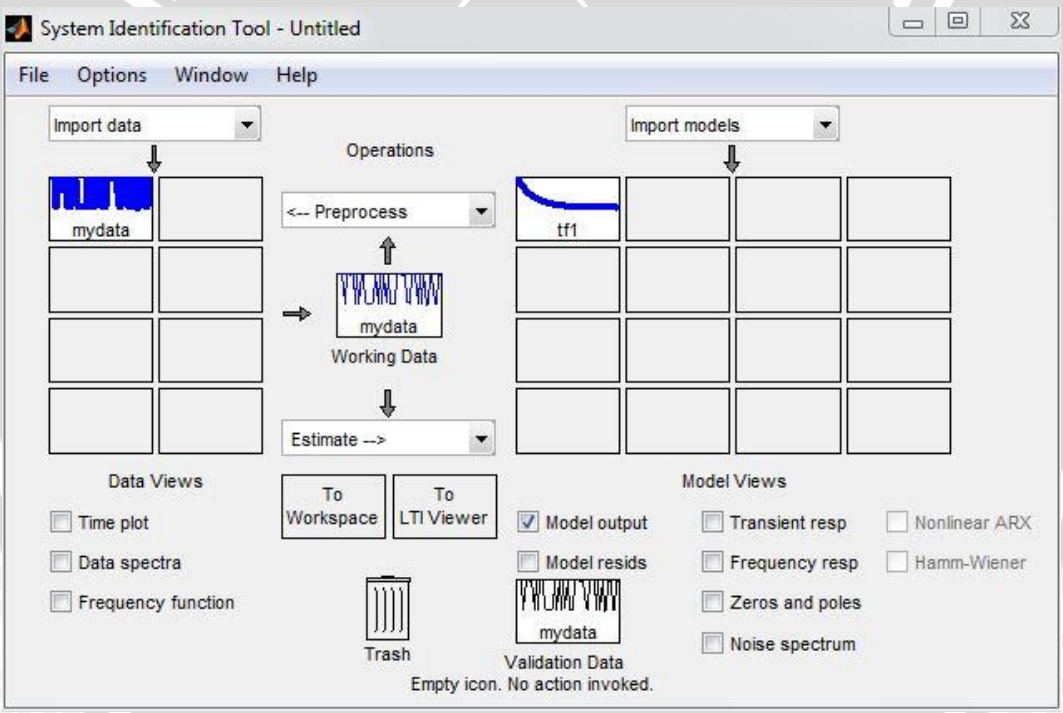


Gambar 3.6 Input PRBS



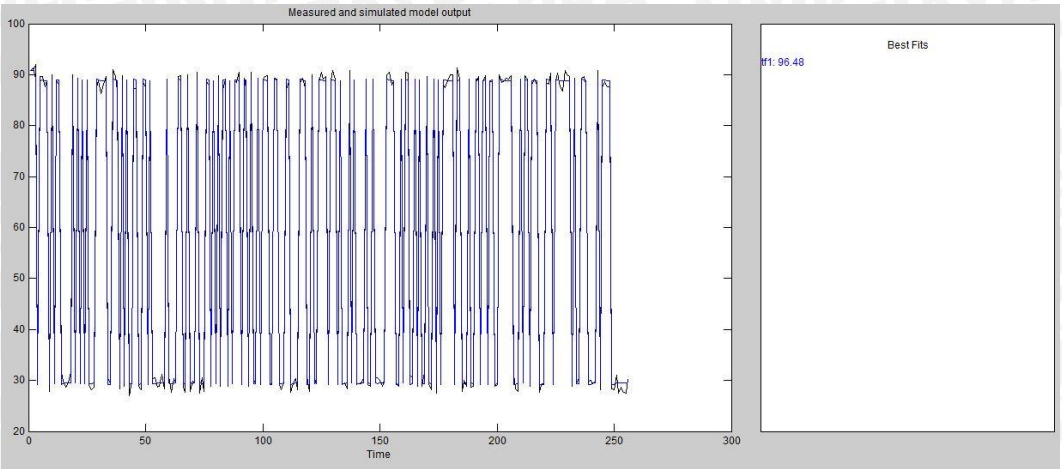
Gambar 3.7 Output PRBS.

4. Selanjutnya identifikasi sistem dilakukan dengan menggunakan Matlab 2012. Data di inport ke blok *System Identification Toolbox*. Gambar tampilan *System Identification Toolbox* dapat dilihat pada Gambar 3.8.



Gambar 3.8 Tampilan System Identification Tool

5. Setelah melakukan beberapa estimasi model dengan beberapa variasi data maka didapatkan fungsi alih dari motor DC servo dengan best fit sebesar 96,48. Hasil estimasi dapat dilihat pada Gambar 3.9.



Gambar 3.9 Hasil Estimasi Model.

6. Setelah melakukan identifikasi system didapatkan fungsi alih sistem sebagai berikut :

$$\frac{U(s)}{E(s)} = \frac{4.145S + 6.505}{S^2 + 5.541S + 6.598} \dots\dots (3-1)$$

$$\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \dots\dots (3-2)$$

$$\omega_n = \sqrt{6.598} = 2.568 \dots\dots (3-3)$$

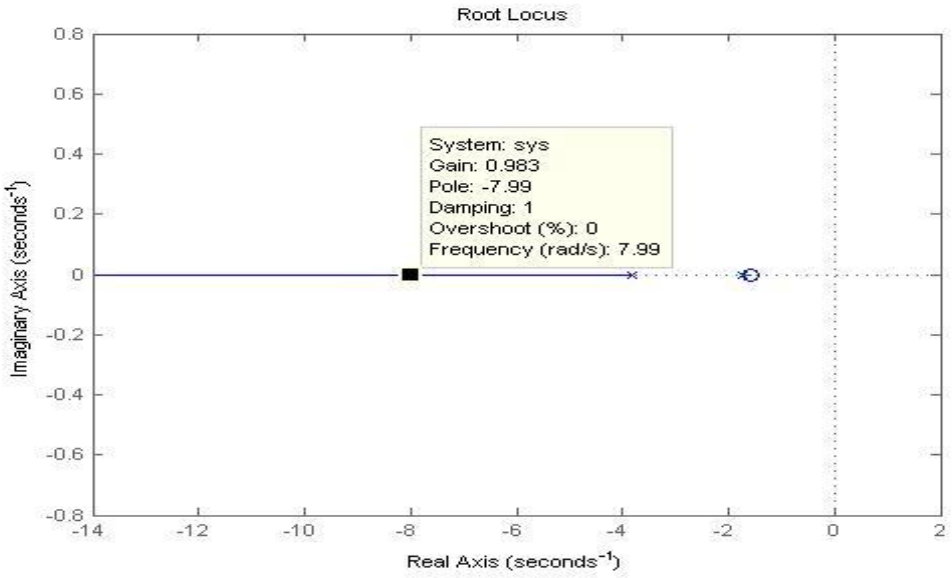
$$2\xi\omega_n = 5.541$$

$$\xi = \frac{5.541}{2\ x\ 2.568} = 1.078 \dots\dots (3-4)$$

3.4.6.4 Penentuan Parameter Kontroler PID dengan Menggunakan Metode *root locus*

Agar robot eco mampu dikontrol dengan baik maka perlu ditambahkan kontroler pada robot eco. Kontroler yang dipilih adalah kontroler Proportional Integral Derivative (PID). Setelah didapatkan fungsi alih sistem yaitu $\frac{Y(s)}{U(s)} = \frac{4.145s+6.505}{s^2+5.541s+6.598}$, selanjutnya adalah menetukan letak*poleloop* tertutup.

Fungsi alih dari sistem diketahui berorde dua. Untuk mendapatkan parameter PID maka tentukan terlebih dahulu letak *pole*pada diagram *root locus*. *Pole* yang digunakan penelitian ini adalah $s_1 = -7.99$. Pada $poles_1$diperoleh nilai gain=0.983, rasio redaman=1, dan *overshoot*=0. Letak *pole* pada root locus dapat dilihat pada Gambar 3.10



Gambar 3.10Letak *Pole* pada diagram *root locus*

Langkah berikutnya yang kita lakukan setalah meletakkan *pole* pada diagram root locus adalah mensubtitusi pole $s_1$kedalam fungsi alih sistem yang telah kita dapatkan tadi, setelah memasukkan nilai $s_1$kedalam fungsi alih maka kita harus menentukan parameter Ki agar nantiya didapatkan parameter Kp, Ki , dan Kd. Mencari parameter Kp, Ki, dan Kd menggunakan Matlab 2012,Berikut adalah listing program yang digunakan pada Matlab 2012 :

```
%Masukkan nilai s1 yang telah ditentukan
s1=-7.99
%Masukkan nilai KI
KI=[2 4 6 8 10]
%Masukkan fungsi alih sistem
plant_num=[0 4.145 6.505];
plant_den=[1 5.541 6.598];

s1mag=abs(s1)
beta= angle(s1)
plant_a1= polyval(plant_num,s1)/polyval(plant_den,s1);
plants1mag=abs(plant_a1)
psi=angle(plant_a1)
t=0:1:20:300;
```

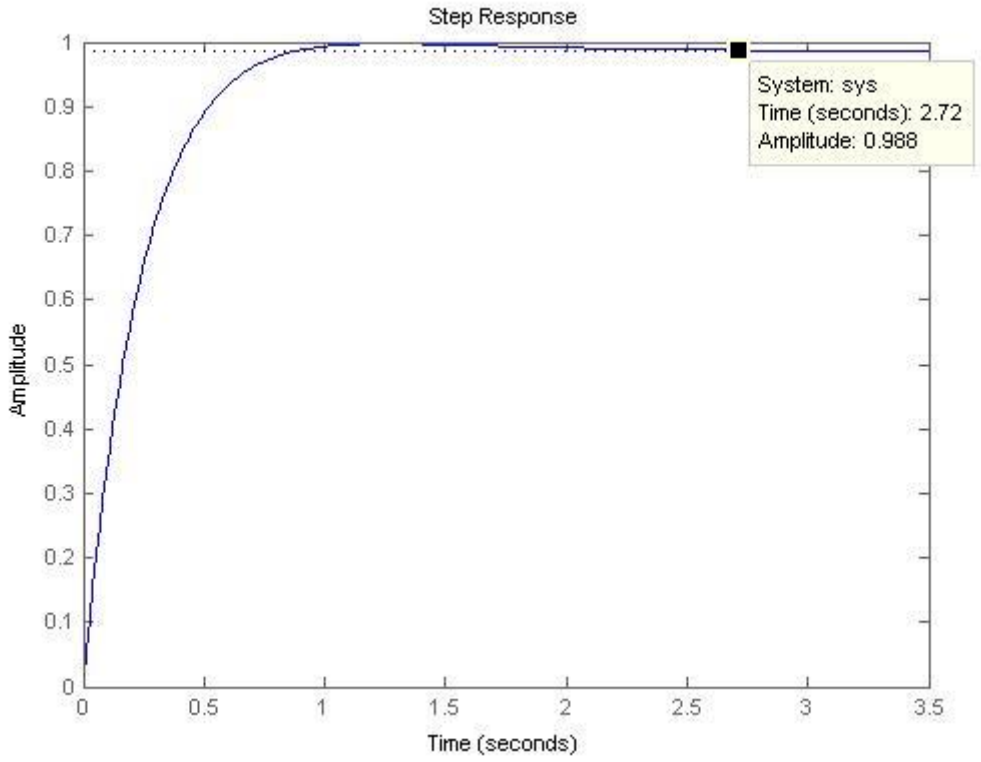
```
for k=1:5
KP=-sin(beta+psi)/(plants1mag*sin(beta))-2*KI(k)*cos(beta)/s1mag
nilai_KI=KI(k)
KD=sin(psi)/(s1mag*plants1mag*sin(beta))+KI(k)/s1mag^2

Gcnum=[KD KP KI(k)];
Gcden=[0 1 0];
Tnum= conv(plant_num,Gcnum);
Tden=conv(plant_den,Gcden)+conv(plant_num,Gcnum);
r=roots(Tden)

step(Tnum,Tden,t)
hold on
end
hold off
figure, rlocus(Tnum,Tden)
```

Setelah melakukan perhitungan menggunakan matlab didapatkan parameter Kp, Ki, dan Kd pada Tabel 3.2

Tabel 3.2 Parameter PID dengan s1=-7.99

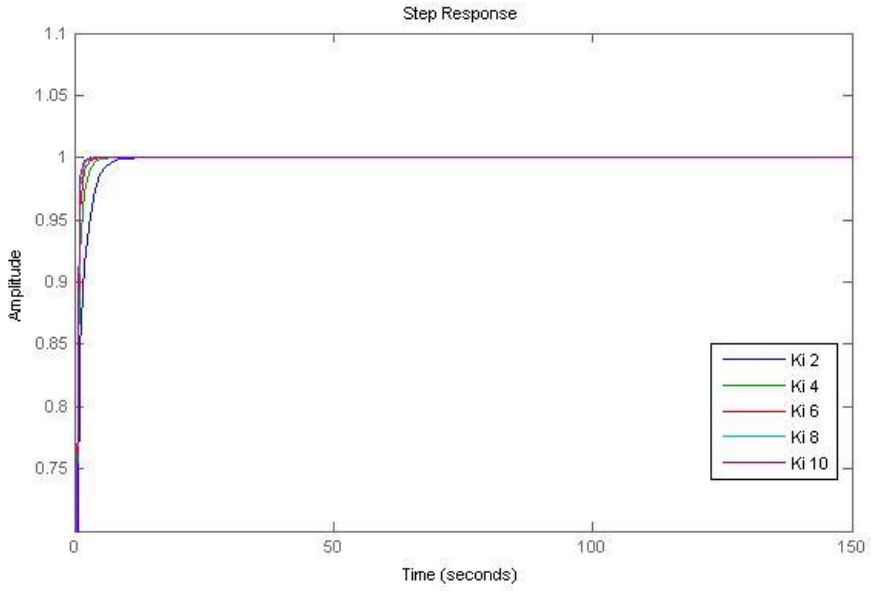
No.	Kp	Ki	Kd	Pole 1	Pole 2	Pole 3
1.	2.4670	2	0.1544	-7.99	-1.6261	-0.6106
2.	2.9676	4	0.1857	-7.99	-1.6763	-1.0978
3.	3.4682	6	0.2170	-7.99	-1.8413	-1.3966
4.	3.9688	8	0.2484	-7.99	-2.14935	-1.4935
5.	4.4695	10	0.2797	-7.99	-2.4745	-1.5237

Setelah didapatkan parameter Kp, Ki, dan Kd maka dilakukan pengujian terhadap sistem dengan nilai parameter yang sudah didapatkan. Pada Gambar 3.11didapatkan grafik *output* sistem ketika tidak menggunakan kontroler PID.



Gambar 3.11Output sistem tanpa kontroler PID dengan *input unit step*

Pada Gambar 3.12berikut ini dapat dilihat ketika sistem menggunakan kontroler PID dengan beberapa parameter yang telah didapatkan dr hasil perhitungan dengan menggunakan Matlab 2012.



Gambar 3.12 Output sistem ketika menggunakan kontroler PID

Pada Gambar 3.11 dapat dilihat respon sistem ketika tidak menggunakan kontroler PID lebih lama untuk mencapai *steady state* dan melebihi setpoint yang di inginkan. Namun dapat dilihat perbedaan pada Gambar 3.12 dimana ketika sistem menggunakan kontroler PID hasil perhitungan dari matlab maka respon sistem lebih cepat mencapai titik *steady state.*

3.4.7. Diskritisasi

Pada perhitungan menggunakan matlab 2012 didapatkan PID terbaik yaitu Ki=4.4695, Ki=10, dan Kd=0.2797. Berikut adalah persamaan kontroler PID dalam bentuk persamaan transformasi laplace.

$$U(S) = \left(Kp + \frac{Ki}{s} + Kds\right)E(s) \dots\dots (3-5)$$

Persamaan tersebut belum bisa dimasukkan pada mikrokontroler atau arduino dikarenakan persamaan tersebut masih dalam bentuk kontiyu, maka dari itu persamaan tersebut harus diubah terlebih dahulu ke dalam bentuk diskrit Dengan cara merubah persamaan tersebut menggunakan transformasi z. Didalam transformasi z dibutuhkan periode sampling (Ts). Metode yang digunakan untuk disktritisasi adalah metode *Backward Difference* dikarenakan metode ini mempunyai sensitifitas yang lebih rendah dari metode yang lainnya sehingga sistem tidak mudah mengalami suatu perubahan atau osilasi saat diberi gangguan dan tetap mempertahankan performansinya. Metode tersebut dilakukan dengan cara mensubstitusi notasi s dengan sebuah persamaan berikut:

$$S = \frac{1 - z^{-1}}{Ts} \dots\dots (3-6)$$

Persamaan tersebut menjadi :

$$U(z) = \left[Kp + \frac{Ki}{\frac{1-z^{-1}}{Ts}} + Kd\left(\frac{1-z^{-1}}{Ts}\right)\right]E(z)$$

$$U(z) = \left[Kp + \frac{Ki}{1-z^{-1}} + \frac{Kd}{Ts}(1-z^{-1})\right]E(z) \dots\dots (3-7)$$

Maka didapatkan persamaan :

- Kontroler Proportional : $U_p(z) = KpE(z) \dots\dots (3-8)$
- Kontroler integral : $U_i(z) = \frac{Ki\,Ts}{1-z^{-1}}E(z)$

$$U_i(z) = [U_i(z)z^{-1} + Ki\,Ts\,E(z)] \dots\dots (3-9)$$

- Kontroler derivative : $U_d(z) = \frac{Kd}{Ts}(1 - z^{-1})E(z)$

$$U_d(z) = \frac{Kd}{Ts}(E(z) - E(z)z^{-1})E(z) \dots\dots (3-10)$$

Dari persamaan diatas diubah menjadi persamaan beda sehingga persamaan tersebut menjadi :

$$C(z) = Kp\ E(k) + \big(Ci(k-1) + Ki\ x\ Ts\ E(k)\big) + (\frac{Kd}{Ts}\big(E(k) - E(k-1)\big)..(3-11)$$

Pada persamaan diatas (k-1) adalah kondisi sebelumnya. Setelah didapatkan persamaan tersebut barulah persamaan bisa dimasukkan kedalam mikrokontroler.

UNIVERSITAS BRAWIJAYA

BAB IV
HASIL DAN PEMBAHASAN

Pada bab ini di jelaskan beberapa percobaan yang telah dilakukan untuk menunjang pengujian secara keseluruhan sistem nantinya. Pengujian dilakukan agar mampu mengetahui karakteristik dan memudahkan pada saat pembuatan robot nantinya. Pengujian dibagi menjadi beberapa, yaitu:

1. Pengujian Motor DC servo Hs-5685 MH
2. Pengujian Sensor garis
3. Pengujian Sistem Secara Keseluruhan

4.1. Pengujian Motor DC Servo

Pengujian Motor DC servo dilakukan untuk mengetahui karakteristik dari motor DC servo HS-5685 itu sendiri. Selain itu pengujian ini juga dilakukan untuk mencari nilai yang nantinya akan dijadikan batas atas dan batas bawah yang akan menjadi masukan dari PRBS (Pseudo Random Binary Sequance). Dengan cara melihat kenaikan yang secara liniear pada Ton (Pulsa High) pada servo. Pengujian ini dilakukan dengan cara memasang PC Lab2000SE pada keluaran data Arduino.

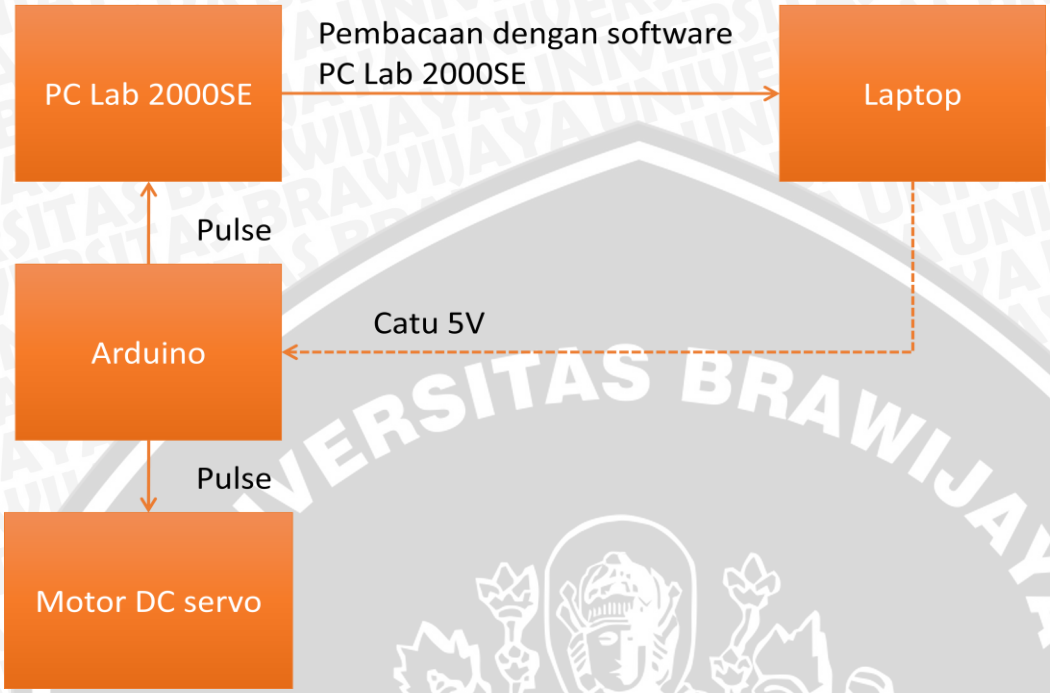
Peralatan yang digunakan pada pengujian motor DC servo terdriri dari :

1. Motor DC servo HS-5685
2. PC Lab2000SE
3. Kabel penghubung PC Lab dan laptop
4. Laptop
5. Arduino mega 2560
6. Software PC Lab 200SE

Prosedur percobaan yang dilakukan adalah sebagai berikut:

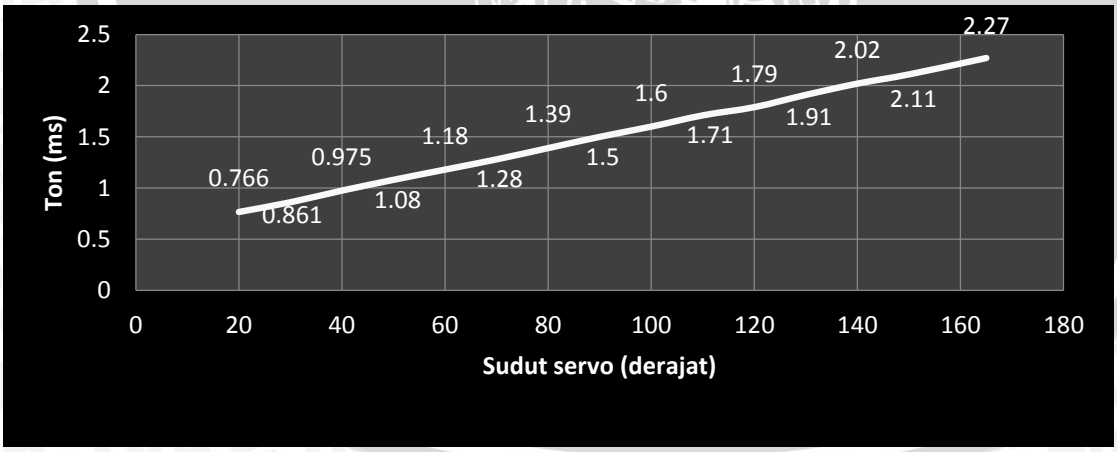
1. Menghubungkan antara PC Lab 2000SE pada chanel 1 dengan keluaran data pada arduino, dan juga ground yang digunakan pada arduino.
2. Menghubungkan PC Lab 2000SE dengan Laptop.
3. Menghubungkan Arduino mega 2560
4. Memasukkan algorithma motor DC servo untuk tiap sudut.
5. Melihat hasil percobaan pada software PC Lab 2000SE

Pada Gambar 4.1 berikut ini adalah Blok pengujian motor DC servo HS-5685 :



Gambar 4.1 Blok diagram pengujian motor DC servo

Hasil dari percobaan motor DC srevo dapat dilihat pada Gambar 4.2 berikut :



Gambar 4.2Hasil Percobaan pengujian Motor DC Servo

Dari hasil pengujian motor DC servo tersebut didapatkan data bahwa servo HS-5685 MH mempunyai rentang sudut 20°-165° dan mempunyai range sebesar 0°-145°. Dari grafik juga dapat dilihat kenaikan Ton(Pulsa High) yang linear berada pada 30°-

90°, sehingga nantinya ketika pengujian PRBS 30° menjadi batas bawah dan 90° menjadi batas atas.

4.2. Pengujian Sensor Garis

Pengujian pada sensor garis dilakukan untuk mengetahui karakteristik dari masing-masing photodioda, sehingga nantinya dapat mempermudah ketika membuat algoritma untuk sistem secara keseluruhan. Pengujian ini dilakukan dengan cara melihat nilai ADC pada sensor garis.

Pengujian sensor garis dilakukan pada semua warna yang berada di lintasan, dengan warna LED terbaik yang telah didapatkan yaitu RED+GREEN. Pada Tabel 4.1 berikut ini adalah pengujian warna yang dilakukan pada warna lapangan putih yang berada di lapangan :

Tabel 4.1 Hasil pengujian pada warna putih

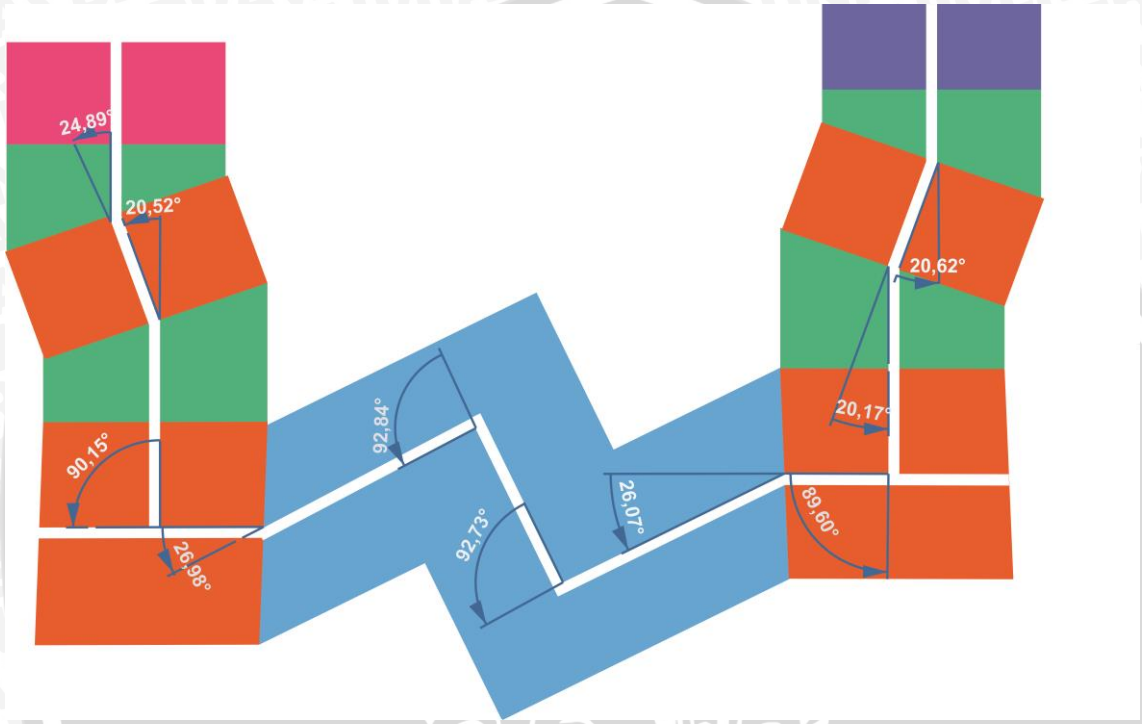
Percobaan ke-	Uji coba					
	Percobaan lapangan putih					
	S1	S2	S3	S4	S5	S6
1	203,32	793,74	226,78	230,69	238,51	215,05
2	207,23	809,65	230,69	234,6	234,6	218,96
3	211,14	789,83	222,87	238,51	230,69	215,05
4	207,23	793,94	226,78	234,6	230,69	215,05
5	203,32	797,65	226,78	230,69	234,6	211,14

Pada pengujian tersebut dapat diketahui bahwa masing-masing sensor garis mempunyai karakteristik masing-masing. Dimana sensor 2 mempunyai perbedaan, ketika sensor 1, sensor 3,sensor 4, sensor 5, dan sensor 6 mempunyai data ADC sekitar 200-240 namun sensor 2 mempunyai data ADC antara 790-810. Sehingga nantinya ketika memprogram robot sensor 2 mempunyai batas threshold yang berbedan daripada sensor lainnya.

Pada Tabel 4.1 sensor 3 dan sensor 4 adalah sensor yang digunakan sebagai setpoint nantinya ketika pengujian sistem secara keseluruhan. Dimana setpoint dan indikasi robot berjalan dengan lurus ketika kedua sensor tersebut mendeteksi warna putih pada lapangan.

4.3. Pengujian Sistem Secara Keseluruhan

Pengujian sistem secara keseluruhan dilakukan untuk mengetahui apakah perancangan berhasil atau tidak dan nantinya dapat ditarik kesimpulan dan saran untuk penelitian selanjutnya. Pengujian sistem secara keseluruhan dilakukan pada lintasan yang menyerupai dengan bentuk lapangan asli pada Kontes Robot ABU Indonesia (KRAI). Bentuk lintasan dapat dilihat pada gambar 4.3 berikut :



Gambar 4.3 Bentuk Lintasan Pengujian

Dari lintasan tersebut dibagi menjadi 4 pengujian sistem dimana masing-masing lintasan mempunyai bentuk lintasan yang berbeda-beda sehingga diharapkan nantinya percobaan dapat ditarik kesimpulan dari masing-masing lintasan. Bentuk lintasan dapat dibagi menjadi 4 yaitu :

1. Lintasan Pengujian A (dengan lintasan 20,62° ke kanan dan 20,17° ke kiri)
2. Lintasan Pengujian B (dengan lintasan 89,60° ke kanan dan 26,07° ke kiri)
3. Lintasan Pengujian C (dengan lintasan 92,73° ke kanan, 92,84° ke kiri dan 26,98° ke kanan)
4. Lintasan Pengujian D (dengan lintasan 90,15° ke kanan dan 20,52° ke kiri, 24,89° ke kanan)

Bentuk lintasan adalah gangguan yang diberikan kepada robot agar nantinya dapat dilihat apakah robot mempunyai performa yang bagus ketika diberi gangguan yang besar ataupun gangguan yang kecil. Semakin curam belokan yang berada di

masing-masing lintasan semakin besar gangguan yang diberikan kepada robot. sehingga masing-masing lintasan mempunyai gangguan yang berbeda.

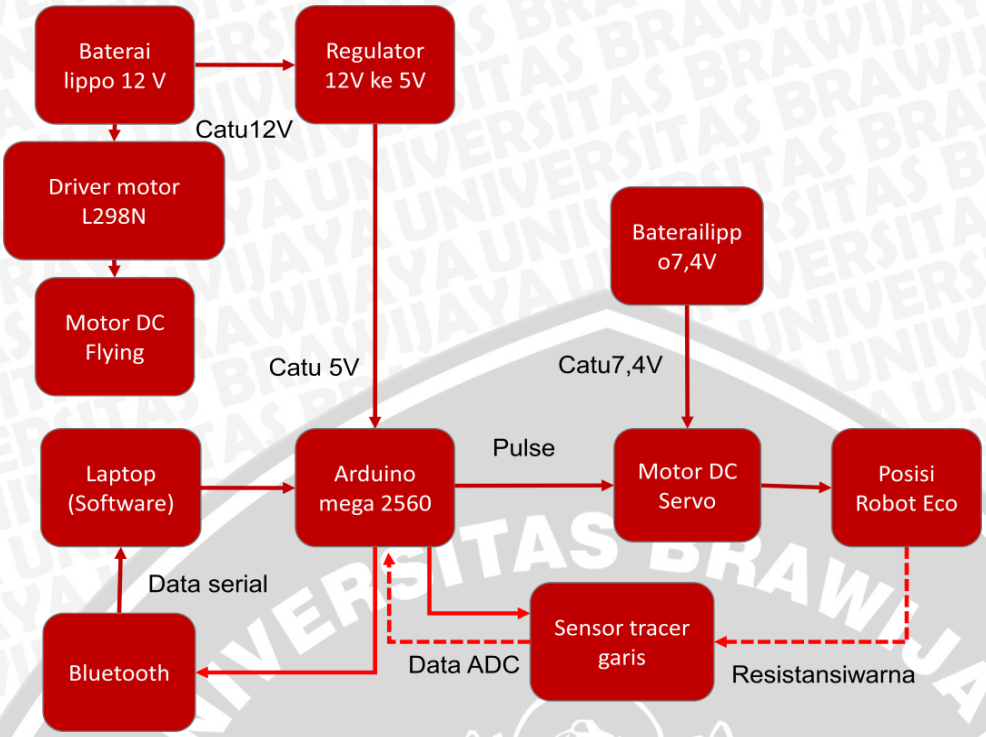
Peralatan yang digunakan pada pengujian sistem secara keseluruhan adalah sebagai berikut :

1. Baterai lippo 12V.
2. Baterai lippo 5V.
3. Arduino mega 2560.
4. Laptop.
5. 2 buah Bluetooth CZ-HC-05.
6. USB TTL.
7. Lapangan Pengujian.
8. Kabel penghubung.
9. Sensor garis**.**

Prosedur percobaan yang dilakukan pada percobaan sistem secara keseluruhan adalah sebagai berikut :

1. Membuka aplikasi Arduino.
2. Memasukkan algorithma sistem secara keseluruhan pada Arduino mega.
3. Membuka aplikasi Tera term untuk melihat data serial.
4. Menaruh robot pada posisi start pada masing-masing lintasan
5. Menghidupkan robot eco dan menunggu bluetooth master dan slave connect.
6. Setelah connect lepas robot dan biarkan berjalan pada lintasan.
7. Melihat data keluaran sensor pada serial monitor pada laptop

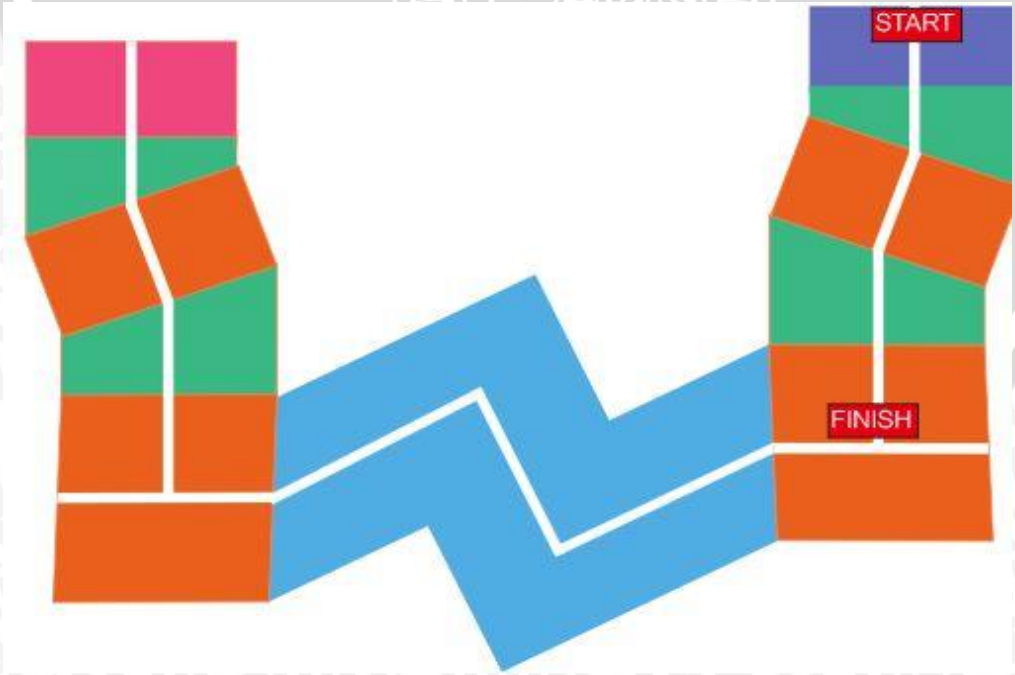
Pada Gambar 4.4 berikut ini adalah blok diagram secara sistem secara keseluruhan keseluruhan :



Gambar 4.4 Blok diagram sistem secara keseluruhan

4.3.2. Pengujian Pada Lintasan A

Pengujian pada lintasan A bertujuan untuk mengetahui apakah ketika menggunakan kontroler PID ataupun tidak menggunakan kontroler PID robot mampu sampai dari garis start sampai garis finis. Lintasan A memberikan gangguan yang kecil, bentuk dari lintasan A dapat dilihat pada gambar 4.5



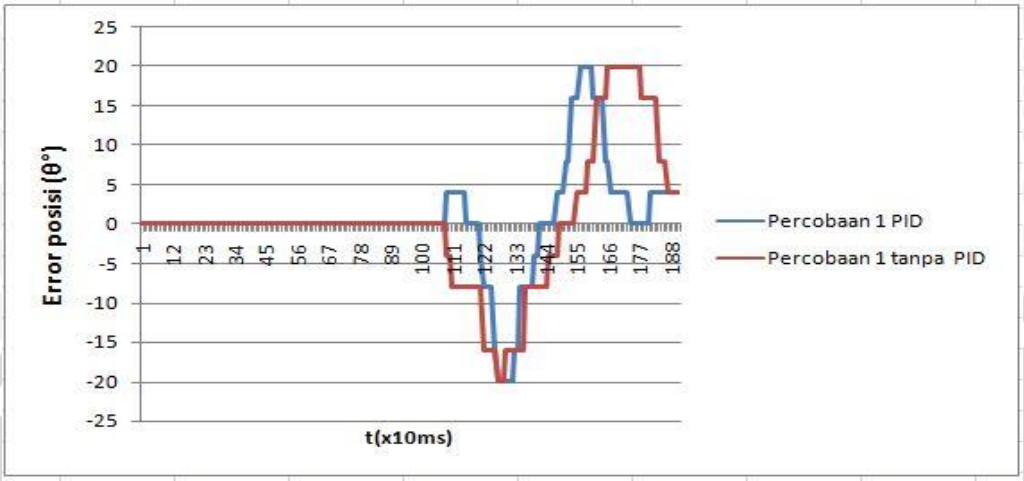
Gambar 4.5 Lintasan Pengujian A

Keterangan

Start : Posisi Awal Robot

Finish : Posisi Akhir Robot

Lama nya percobaan pada lintasan A ketika diukur menggunakan stopwatch adalah sekitar +- 2 detik Hasil percobaan dari lintasan A dapat dilihat pada gambar 4.6



Gambar 4.6 Grafik pengujian pada Lintasan A.

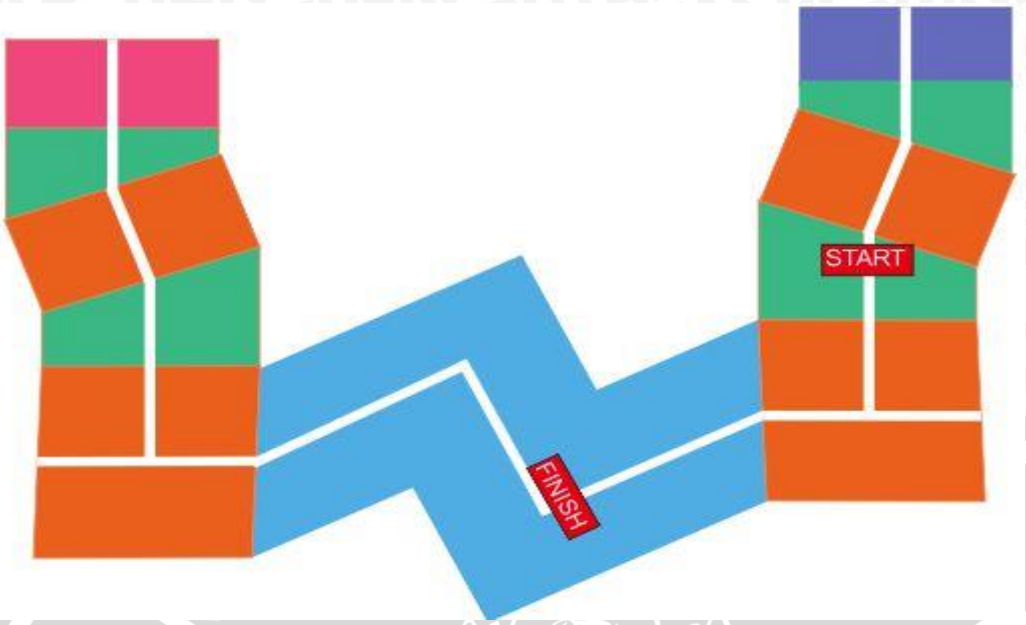
Dapat dilihat pada gambar 4.6 robot ketika menggunakan kontroler PID mempunyai error yang lebih sedikit daripada ketika tidak menggunakan kontroler PID. Namun perbedaan tidak begitu signifikan ketika pada pengujian pada lintasan A, dengan persentase keberhasilan dari start sampai finis ketika menggunakan kontroler PID adalah 100% dan ketika tidak menggunakan kontroler PID adalah 100%. Error posisi rata-rata yang dihasilkan ketika menggunakan kontroler PID adalah 3,17° dan ketika robot tidak menggunakan kontroler PID adalah 4,75°.

Pada gambar 4.6 grafik menunjukkan ketika t=1090 ms(t=1,09s) robot berbelok ke kanan sampai dengan t=1490ms (t=1,49s) dan ketika t=1560ms (t=1,56s) robot ber belok ke kiri sampai dengan t=1820ms (t=1,82s) lalu robot bergerak lurus. Semakin negatif error posisi pada grafik menunjukkan robot semakin berbelok kekanan, sedangkan ketika semakin positif error posisi pada grafik menunjukkan robot semakin berbelok ke kiri.

4.3.3. Pengujian Pada Lintasan B

Pengujian pada lintasan B dilakukan dengan cara membandingkan antara percobaan menggunakan kontroler PID dan ketika robot tidak menggunakan kontroler PID. Percobaan ini relatif lebih besar gangguan nya daripada lintasan A dikarenakan

terdapat lintasan yang cukup curam 90° ke arah kanan. Bentuk lintasan B dapat dilihat pada gambar 4.7



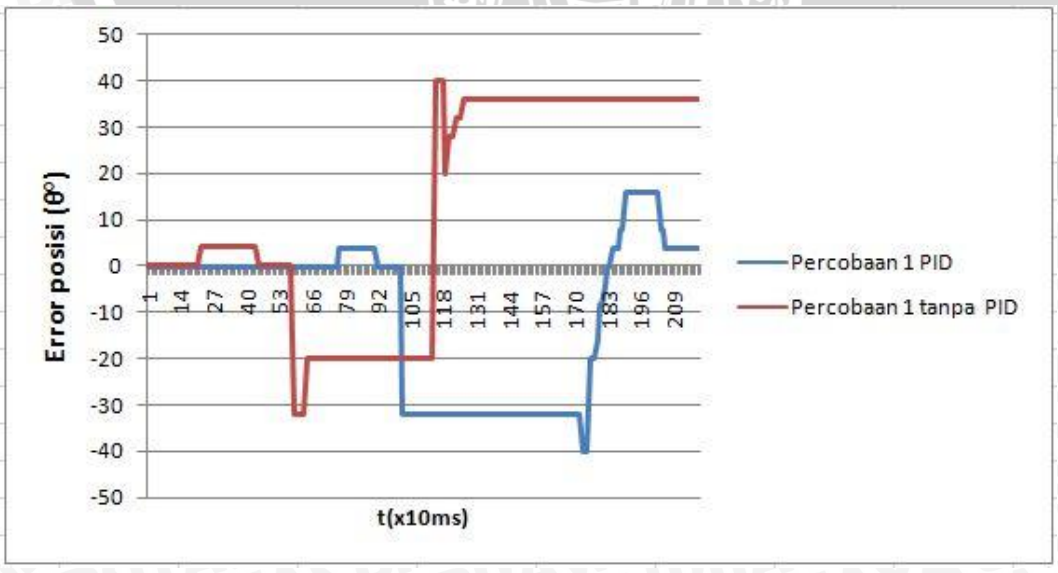
Gambar 4.7 Lintasan Pengujian B

Keterangan

Start : Posisi Awal Robot

Finish : Posisi Akhir Robot

Lama nya percobaan pada lintasan B saat posisi robot dari start sampai finish ketika diukur menggunakan stopwatch adalah +-2 detik. Hasil pengujian pada lintasan B dapat dilihat pada gambar 4.8 berikut



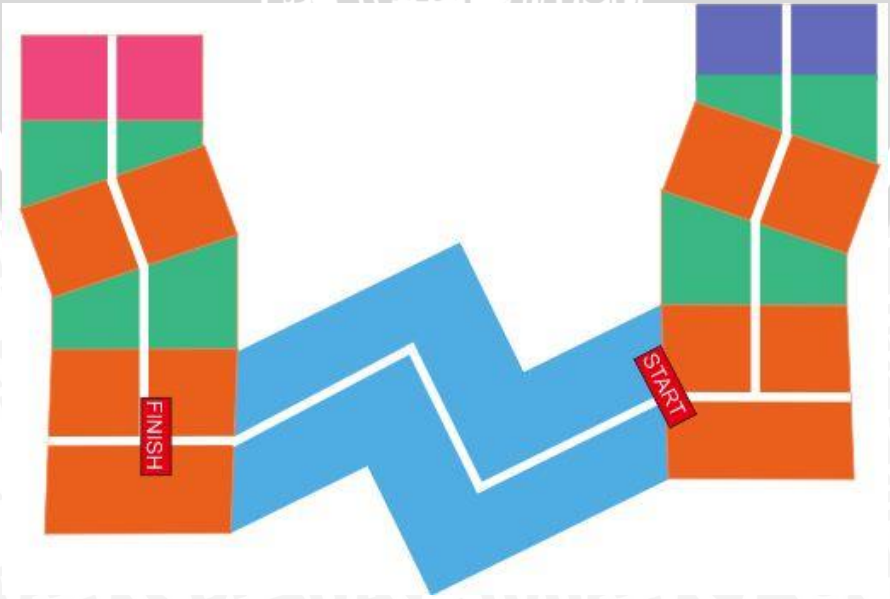
Gambar 4.8 Grafik Pengujian Pada Lintasan B

Dari beberapa kali percobaan pada lintasan B didapatkan hasil ketika menggunakan kontroler PID tingkat keberhasilannya adalah 80% dan ketika tidak menggunakan kontroler PID tingkat keberhasilannya adalah 40%. Dengan error posisi rata-rata ketika menggunakan kontroler PID adalah 13,05° dan ketika tidak menggunakan kontroler PID error posisi rata-ratanya adalah 22,8°.

Pada gambar 4.8 ketika menggunakan kontroler PID saat t=1030ms (t=1,03s) robot berbelok kearah kanan sampai t=1800ms (t=1,8s) setalah itu barulah robot berbelok ke arah kiri pada saat t=1910ms (t=1,91s) sampai dengan ketika t=2070ms (t=2,07s). Dan ketika robot tidak menggunakan kontroler PID pada awalnya robot mampu belok kanan namun pada saat t=1160ms (t=1,160s) robot keluar lintasan dan membaca lapangan putih yang berada diluar lintasan. Pada grafik juga dapat kita tarik kesimpulan ketika lapangan mempunyai belokan semakin tajam maka error posisi dari robot juga akan semakin besar. Hal tersebut dapat dilihat ketika robot menggunakan kontroler PID pada saat belok kekanan sebesar 90° dan ketika belok kekiri 60°. Lamanya robot berbelok juga menjadi kekurangan pada bentuk mekanik dari robot.

4.3.4. Pengujian Pada Lintasan C

Pada pegujian lintasan C diharapkan robot mampu melewati lintasan yang berbentuk zigzag, dimana lintasan ini mempunyai gangguan yang paling besar dari pada bentuk lintasan yang lainnya. Dimana robot akan melewati lintasan dengan berbelok ke kanan 90° , berbelok kekiri 90° lalu akan berbelok 60° ke kanan. Bentuk lintasan pada lintasan C dapat dilihat pada gambar 4.9



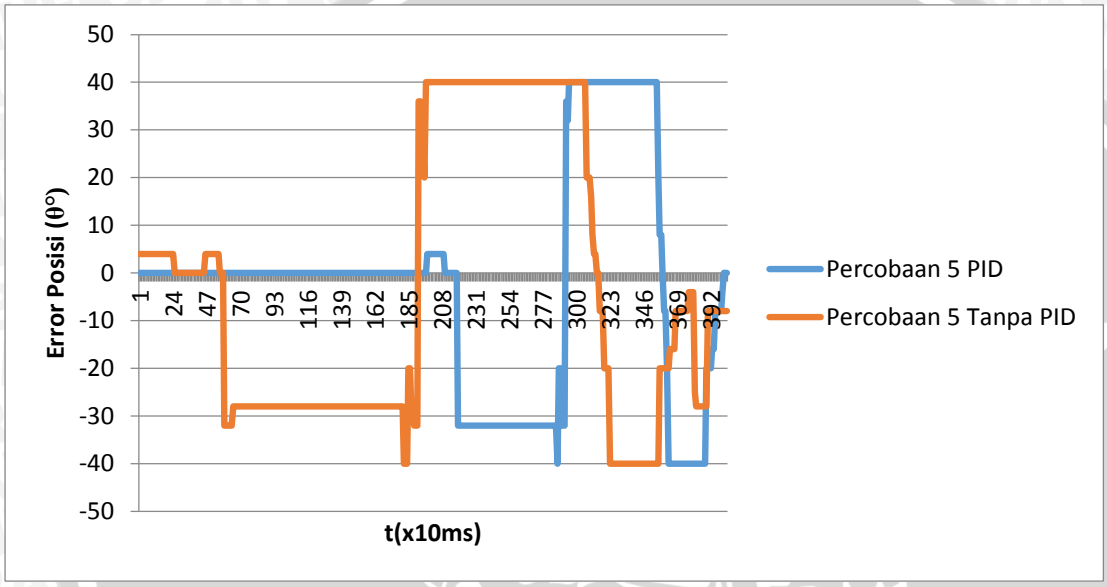
Gambar 4.9 Lintasan Pengujian C

Keterangan

Start : Posisi Awal robot

Finish : Posisi Akhir robot

Lamanya percobaan pada lintasan C dari posisi start sampai finish ketika diukur menggunakan stopwatch adalah sekitar +- 4 detik. Hasil percobaan pada lintasan pengujian C dapat dilihat pada gambar 4.10 berikut:



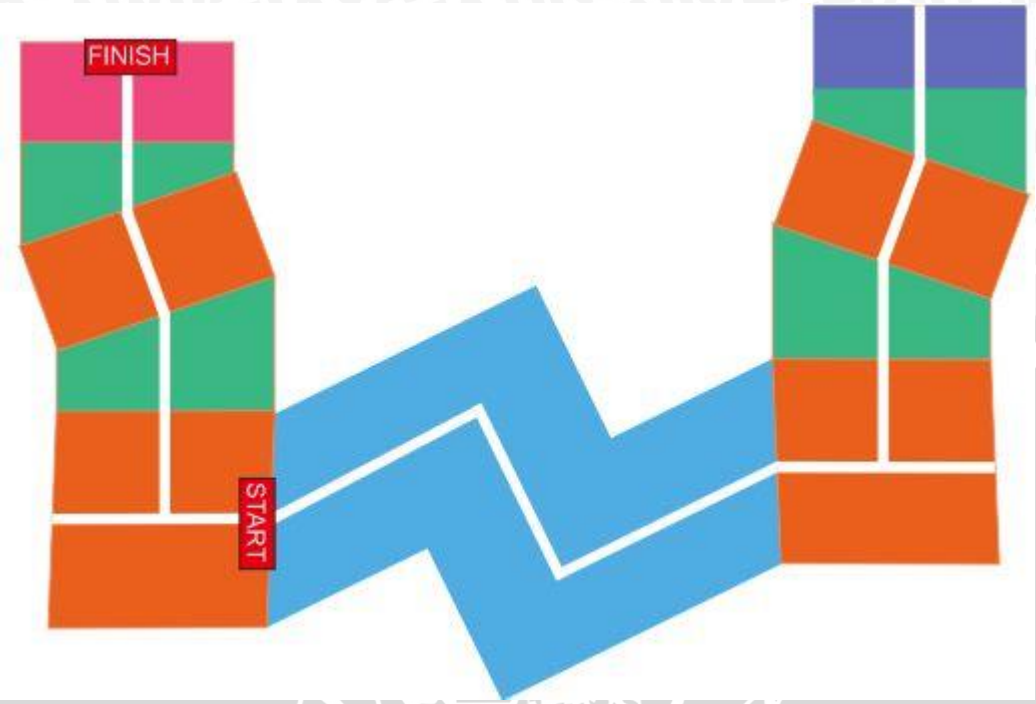
Gambar 4.10 Grafik pengujian Pada Lintasan C

Dari hasil pengujian pada lintasan C didapatkan hasil ketika menggunakan kontroler PID robot mempunyai tingkat keberhasilan 100% dan ketika tidak menggunakan kontroler PID tingkat keberhasilan robot mencapai finis adalah100%. Error posisi rata-rata yang dihasilkan ketika menggunakan kontroler PID adalah 15,25°, sementara ketika menggunakan tidak menggunakan kontroler adalah 26,36°.

Pada gambar 4.10 robot mampu menyelesaikan misi sampai finis ketika menggunakan kontroler PID maupun ketika tidak menggunakan kontroler PID , perbedaan terletak pada banyak nya error ketika tidak menggunakan kontroler PID. Ketika tidak menggunakan kontroler PID robot mampu berbelok sebesar 90° kekanan pada saat t=610ms (t=0,61s) sampai dengan t=1930(t=1,93s) , dan berbelok kiri pada saat t=1940ms(t=1,94s) sampai dengan t=3150ms (t=3,15s), setelah itu robot berbelok kekiri lagi pada saat t=3180ms (3,18s) sampai dengan t=3690ms (t=3,69s).

4.3.5. Pengujian Pada Lintasan D

Pada pengujian lintasan D gangguan yang diberikan relatif sedang dibandingkan dengan lintasan lainnya. Pada lintasan ini kombinasi dari gangguan yang besar dan gangguan yang relatif kecil diberikan sehingga nantinya dapat ditarik kesimpulan. Bentuk lintasan D dapat dilihat pada gambar 4.11.



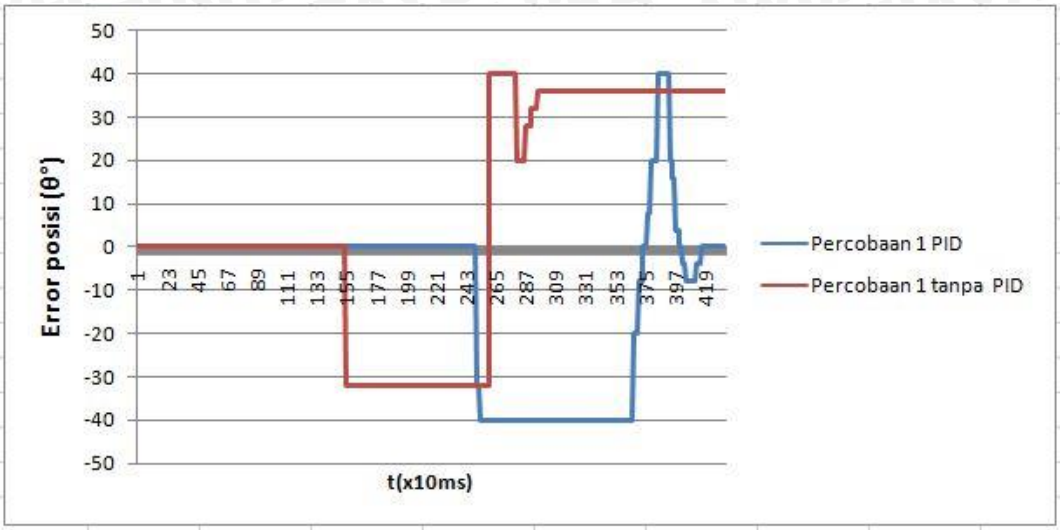
Gambar 4.11 Lintasan Pengujian D.

Keterangan

Start : Posisi Awal Robot

Finish : Posisi Akhir Robot

Lamanya percobaan lintasan D dari posisi start sampai finish ketika menggunakan stopwatch adalah +- 4 detik. Hasil percobaan yang dilakukan pada lintasanD dapat dilihat pada gambar 4.12 berikut :



Gambar 4.12 Grafik Pengujian pada Lintasan D.

Dari hasil pengujian pada lintasan D yang dilakukan beberapa kali diperoleh hasil bahwa ketika robot menggunakan kontroler PID tingkat keberhasilan yang diperoleh untuk mencapai finish adalah 80% dan ketika robot tidak menggunakan kontroler PID maka tingkat keberhasilannya adalah 20%. Error posisi rata-rata robot ketika menggunakan kontroler PID adalah 12,35° dan ketika robot tidak menggunakan kontroler PID error posisi rata-rata yang dihasilkan adalah 22,03°.

Pada gambar 4.12 pada dapat dilihat bahwa robot ketika menggunakan kontroler PID mampu melewati lintasan D dengan baik dan mampu menyelesaikannya dari start sampai dengan finish. Namun ketika robot tidak menggunakan kontroler PID pada saat belokan 90° pertama robot bisa belok namun melebar dan keluar dari lintasan pengujian sehingga robot tidak mampu mencapai finish. Pada gambar 5.9 robot belok ke kanan pada saat t=2520ms (t=2,52s) sampai t=3710ms (t=3,71s). Lalu robot berbelok ke kanan pada saat t=3810 (t=3,81s) sampai dengan t=4040ms (t=4,04s), lalu berbelok lagi ke kiri pada t=4040ms (t=4,04s) sampai dengan t=4170 (t=4,17s) dan akhir nya robot kembali ke posisi lurus. Namun ketika robot tidak menggunakan kontroler PID robot berbelok pada t=1560ms (t=1,56s) sampai t=2620ms (t=2,62s) robot keluar lintasan karena mendeteksi lapangan warna putih yang berada pada lintasan luar.

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan rumusan masalah, perancangan alat dan hasil pengujian yang telah dilakukan dapat diambil beberapa kesimpulan yang nantinya dapat dijadikan pertimbangan nantinya . berikut ini adalah beberapa kesimpulan :

1. Setelah mengidentifikasi sistem didapatkan fungsi alih sistem sebagai berikut : $\frac{Y(s)}{U(s)} = \frac{4.145s+6.505}{s^2+5.541s+6.598}$ dari fungsi alih tersebut didapatkan parameter kontroler PID dengan menggunakan metode root locus. Diperoleh nilai pole sebesar s=-7,99. Parameter Kp=4,4695 , Ki=10, Kd=0,2797.
2. Pada pengujian lintasan B robot ketika menggunakan kontroler PID mempunyai tingkat keberhasilan 80% dan ketika robot tidak menggunakan kontroler tingkat keberhasilan mencapai finish adalah 40%. Error posisi rata-rata ketika menggunakan kontroler PID adalah 13,05° dan ketika robot tidak menggunakan kontroler PID error posisi rata-rata yang dihasilkan adalah 22,8°.
3. Error pada robot eco akan semakin besar apabila sudut pada belokan semakin tajam. Pembacaan data error menunjukkan apabila semakin negatif error yang dihasilkan maka robot akan semakin berbelok kekanan, dan apabila robot semakin positif error maka akan semakin berbelok ke kiri.

5.2. Saran

Berikut ini adalah beberapa hal yang mungkin bisa dijadikan pertimbangan untuk penelitian selanjutnya :

1. Agar data lebih presisi perbanyak sensor garis sehinggaterdapat lebih banyak kombinasi yang bisa digunakan dan pergerakan robot lebih smooth.
2. Untuk aktuator yang digunakan bisa menggunakan servo dynamixel tipe MX yang mampu bergerak 360°. Sehingga range sudut yang di inginkan bisa lebih besar.
3. Penggunaan kontroler lain mungkin bisa menigkatkan performa dari robot.

DAFTAR PUSTAKA

Bolton, W. 1999. *Electronic control systems in mechanical and electrical engineering 2nd edition.* New york: Addison wesley Longman.

Setiawan, I 2008. *Kontrol PID untuk Proses Industri.* Jakarta: Elex Media Komputindo

Mupasanta,S. 2016. *PENGONTROLAN KADAR KEASAMAAN (pH) DAN ALIRAN AIR PADA SISTEM HIDROPONIKNSTROBERI MENGGUNAKAN KONTROLER PID*. Laporan Skripsi . Teknik Elektro. Universitas Brawijaya Malang.

Anonim. 2016. Arduino Mega 2560. diakses dari https://www.arduino.cc/en/Main/ArduinoBoardMega2560. (Pada tanggal 1 juli 2016)

Ogata, K. 1997. *Modern Control Engineering.* New Jersey: Prantice Hall

Anonim. http://e-belajarelektronika.com/definisi-dan-bagian-robot-line-follower/ (Pada tanggal 28 juli 2016)

Fahmizal. 2016.https://fahmizaleeits.wordpress.com/tag/cara-kerja-sensor-garis/ (Pada tanggal 28 juli 2016)

Prabarianto, B. 2016.*Metode Root Locus Untuk Mencari Prameter PID pada pengontrolan Kecepatan Motor DC D-6759 Menggunakan Arduino Mega 2560.* Skripsi. Teknik Elektro . Universitas Brawijaya. Malang.

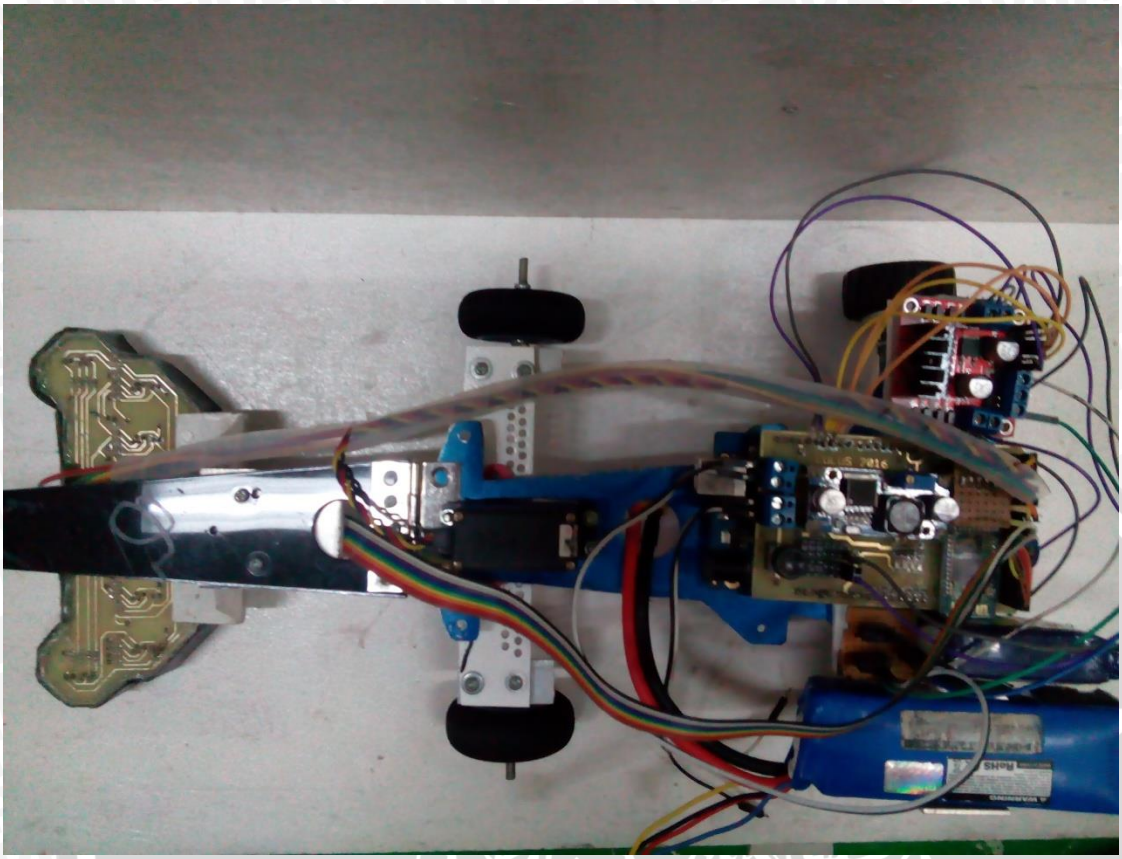
Laksono,Edi. 1995. TEKNIK KONTROL AUTOMATIK (SISTEM PENGATURAN) Jilid 1. Bandung : Penerbit Erlangga.

ABU Robocon. 2016. *ABU Asia-Pasific Robot Contest 2016 Bangkok.* Bangkok: Host Organizing Comittee.

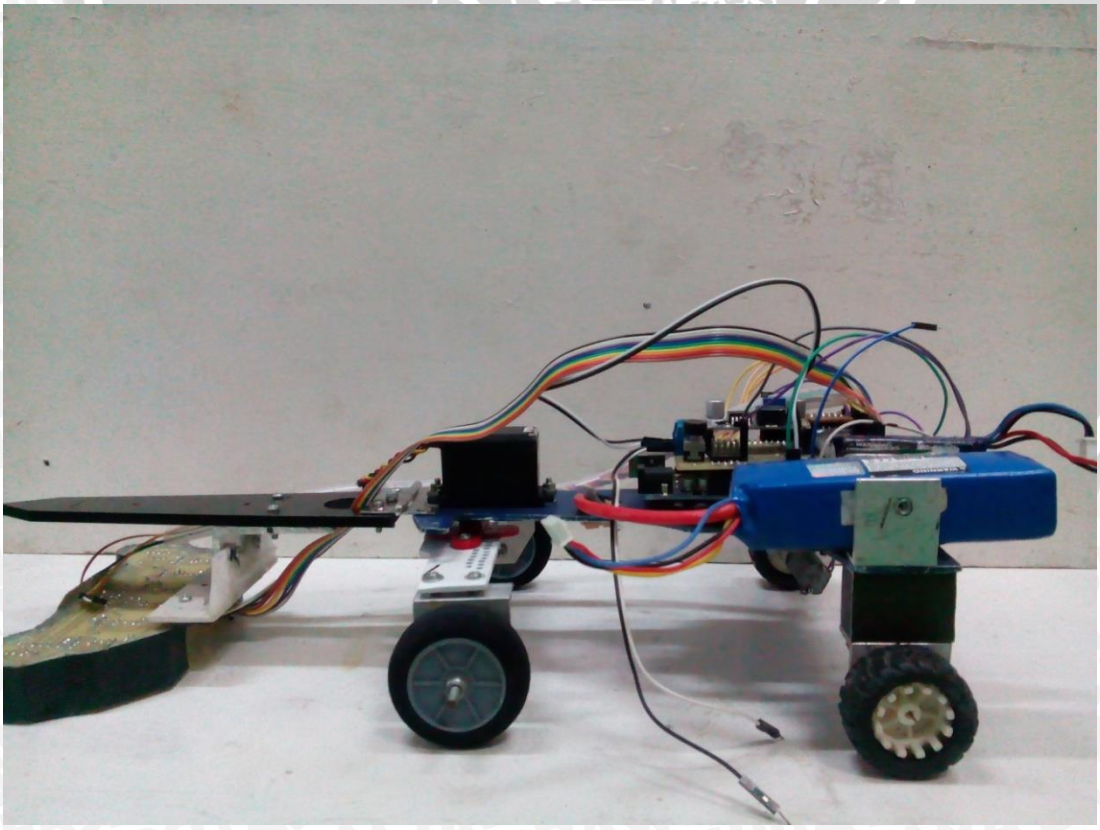
LAMPIRAN I

FOTO ALAT

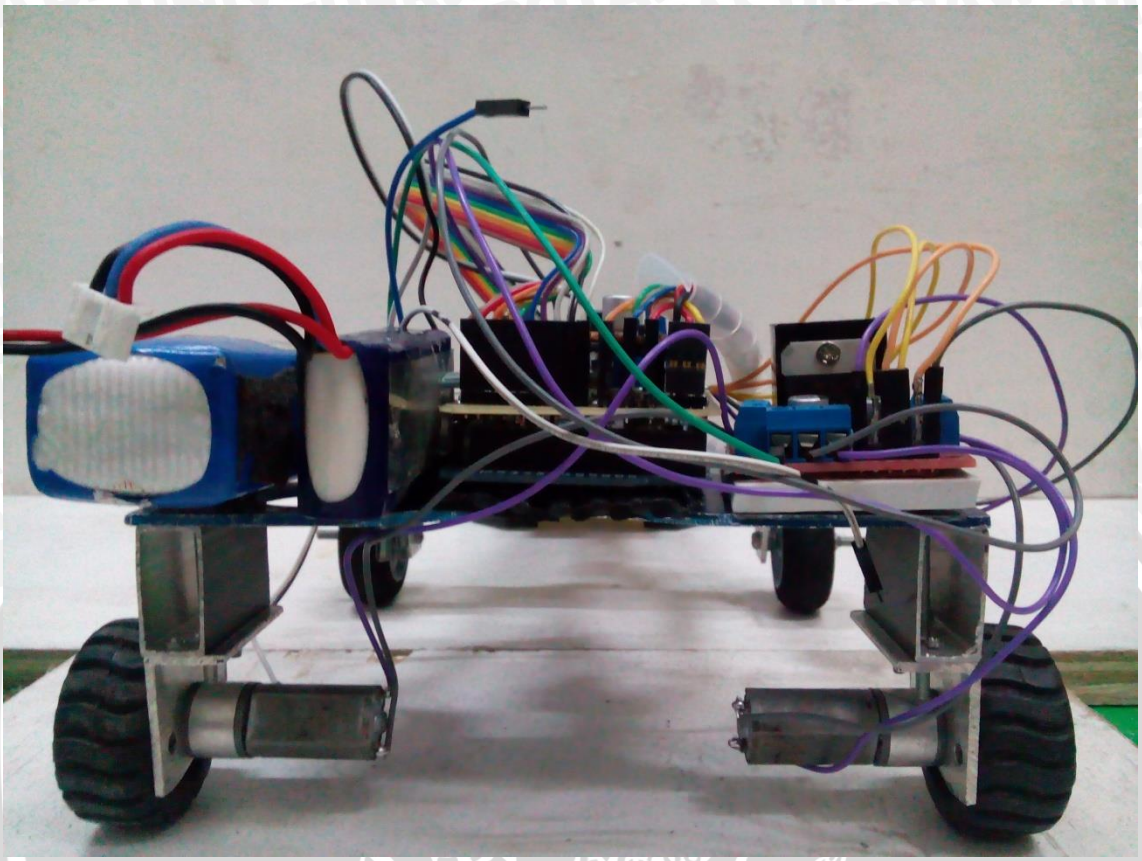
Robot tampak atas



Robot tampak samping



Robot tampak belakang



LAMPIRAN II

LISTING PROGRAM

Listing Program Matlab untuk mencari s1

```
num=[ 0 4.145 6.505]
den=[1 5.541 6.598]

step(num,den)
figure,rlocus(num,den)
```

Listing program matlab untuk mencari parameter kontroler PID dengan *root locus*

```
s1=-7.99
%Masukkan nilai KI
KI=[2 4 6 8 10]
%Masukkan fungsi alih sistem
plant_num=[0 4.145 6.505];
plant_den=[1 5.541 6.598];

s1mag=abs(s1)
beta= angle(s1)
plant_a1= polyval(plant_num,s1)/polyval(plant_den,s1);
plants1mag=abs(plant_a1)
psi=angle(plant_a1)
t=0:1:20:300;

for k=1:5
KP=-sin(beta+psi)/(plants1mag*sin(beta))-2*KI(k)*cos(beta)/s1mag
nilai_KI=KI(k)
KD=sin(psi)/(s1mag*plants1mag*sin(beta))+KI(k)/s1mag^2

Gcnum=[KD KP KI(k)];
Gcden=[0 1 0];
Tnum= conv(plant_num,Gcnum);
Tden=conv(plant_den,Gcden)+conv(plant_num,Gcnum);
r=roots(Tden)
```

```
step(Tnum,Tden,t)
hold on
end
hold off
figure, rlocus(Tnum,Tden)
```

Listing program sistem keseluruhan dari Arduino Mega 2560 ketika menggunakan kontroler PID

```
#include <Servo.h>
Servo bipedal;
int batas_hold[7]={1000,1000,1100,1100,1100,1000,1000};
int sensor_fikrul[7], sensor_line=0,iterasi=1;
int sensor_garis=0,PV=0;
float baca_adc[7];
float Prop,Integ,Diff,MV,error,last1_error,last2_error ;

int Kp=4.4695, Ki=10, Kd=0.2797, Ts=0.01 ;
int sp=0, servo, sudutmin=20, sudutmax=165;
int pwm_kiri=0,pwm_kanan=0, z=0, k=0;

void setup()
{
//unsigned long waktu;
//waktu=micros();

Serial.begin(9600);
bipedal.attach(2);
pinMode (A0, INPUT);
pinMode (A1, INPUT);
pinMode (A2, INPUT);
pinMode (A3, INPUT);
pinMode (A4, INPUT);
pinMode (A5, INPUT);
```

```
pinMode (A6, INPUT);
pinMode (A7, INPUT);
pinMode (A13, OUTPUT);
pinMode (A12, OUTPUT);
pinMode (A11, OUTPUT);
pinMode (A10, OUTPUT);
 // put your setup code here, to run once:
analogWrite (A8, 255); //red depan
analogWrite (A9, 255); //red
analogWrite (A10, 255);//green depan
analogWrite (A11, 255);//green
analogWrite (A12, 255); // blue depan
analogWrite (A13, 255); //blue
//Serial.print("estimasi waktu \t ");
//Serial.println(waktu);
}

void loop()
{

motor_dc (100);
baca_sensor();
PID();
Serial.println(k);delay (10);
}

void baca_sensor(void)
{
 for(iterasi=1;iterasi<7;iterasi++)
 {
  baca_adc[iterasi]=(analogRead(iterasi)*3.91);
  if (baca_adc[iterasi]>batas_hold[iterasi])
  sensor_fikrul[iterasi]=1;
  else
```

```
    sensor_fikrul[iterasi]=0;
//    Serial.print(iterasi);
   }
//  Serial.print(baca_adc[1]); Serial.print("\t");
//  Serial.print(baca_adc[2]); Serial.print("\t");
//  Serial.print(baca_adc[3]); Serial.print("\t");
//  Serial.print(baca_adc[4]); Serial.print("\t");
//  Serial.print(baca_adc[5]); Serial.print("\t");
//  Serial.println(baca_adc[6]); //Serial.print("\r");
//
//  Serial.print(sensor_fikrul[1]); Serial.print("\t");
//  Serial.print(sensor_fikrul[2]); Serial.print("\t");
//  Serial.print(sensor_fikrul[3]); Serial.print("\t");
//  Serial.print(sensor_fikrul[4]); Serial.print("\t");
//  Serial.print(sensor_fikrul[5]); Serial.print("\t");
//  Serial.println(sensor_fikrul[6]);

  sensor_garis =
((sensor_fikrul[6]*32)+(sensor_fikrul[5]*16)+(sensor_fikrul[4]*8)+(sensor_fikrul[3]*4
)+(sensor_fikrul[2]*2)+(sensor_fikrul[1]*1));
  //Serial.println(sensor_garis); Serial.print("\t");

  switch (sensor_garis)
  {
    case 0b110011: PV=0; k=0; break;

    case 0b111011: PV=10 ; k=99-95; break;   //error garis kanan
    case 0b111001: PV=20 ; k=103-95; break;
    case 0b110001: PV=30 ; k=107-95; break;
    case 0b111101: PV=40 ; k=111-95; break;
    case 0b111100: PV=50 ; k=115-95; break;
    case 0b111010: PV=60 ; k=120-95; break;
    case 0b111000: PV=70 ; k=123-95; break;
    case 0b110000: PV=80 ; k=127-95; break;
```

```
    case 0b100000: PV=90 ; k=131-95; break;
    case 0b111110: PV=100 ; k=135-95; break;


    case 0b110111: PV=-10 ; k=91-95; break;     //error garis kiri
    case 0b100111: PV=-20 ;k=87-95; break;
    case 0b100011: PV=-30 ; k=83-95; break;
    case 0b101111: PV=-40 ;k=79-95; break;
    case 0b001111: PV=-50 ;k=75-95; break;
    case 0b010111: PV=-60 ;k=70-95; break;
    case 0b000111: PV=-70 ;k=67-95; break;
    case 0b000011: PV=-80 ;k=63-95; break;
    case 0b000001: PV=-90 ;k=59-95; break;
    case 0b011111: PV=-100 ;k=55-95; break;

    //case 0b000000:  break; motor_dc (0); Serial.println("STOP");break;
   }
}
void PID()
{
  error = sp-PV;
  Prop = Kp * error;
  Integ = (Ki *Ts* (error + last1_error));
  Diff = ((Kd/Ts) * (error - last1_error));
  MV = (Prop + Integ + Diff) *0.1;
  last1_error = error;
  z=95-MV;
  bipedal.write(z);


  delay(10);
}

void motor_dc (int duty)
```

```
{
  pwm_kiri=180*duty/100;
  pwm_kanan=255*duty/100;
  analogWrite (12, pwm_kiri);
  digitalWrite (10, HIGH);
  digitalWrite (11,LOW);
  analogWrite (13, pwm_kanan);
  digitalWrite (10, HIGH);
  digitalWrite (11,LOW);
}
```

Listing program sistem keseluruhan dari Arduino Mega 2560 ketika tidak menggunakan kontroler PID

```
#include <Servo.h>
Servo bipedal;
int batas_hold[7]={1000,1000,1100,1000,1000,1000,1000};
int sensor_fikrul[7], sensor_line=0,iterasi=1;
int sensor_garis=0,PV=0,pwm_kiri=0,pwm_kanan=0, k=0;
float baca_adc[7];

void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
bipedal.attach(2);
pinMode (A0, INPUT);
pinMode (A1, INPUT);
pinMode (A2, INPUT);
pinMode (A3, INPUT);
pinMode (A4, INPUT);
pinMode (A5, INPUT);
pinMode (A6, INPUT);
pinMode (A7, INPUT);
pinMode (A13, OUTPUT);
```

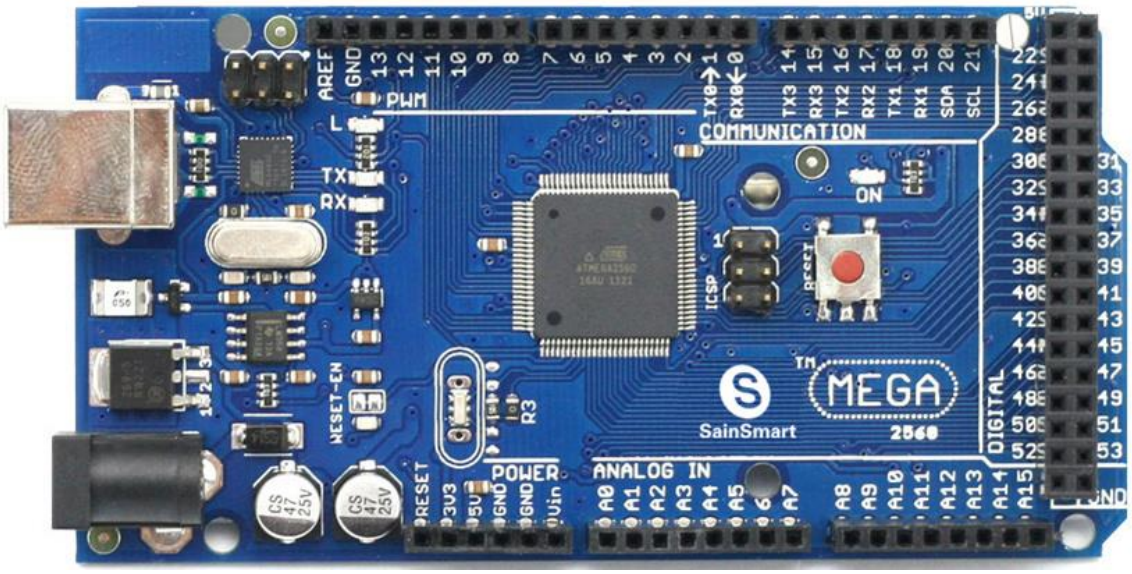
```
pinMode (A12, OUTPUT);
pinMode (A11, OUTPUT);
pinMode (A10, OUTPUT);

analogWrite (A8, 255); //red depan
analogWrite (A9, 255); //red
analogWrite (A10, 255);//green depan
analogWrite (A11, 255);//green
analogWrite (A12, 255); // blue depan
analogWrite (A13, 255); //blue
}

void loop() {
 // put your main code here, to run repeatedly:
motor_dc(100);
baca_sensor();
//Serial.println(PV);
Serial.println(k); delay (10);
}

void baca_sensor(void)
{
 for(iterasi=1;iterasi<7;iterasi++)
 {
  baca_adc[iterasi]=(analogRead(iterasi)*3.91);
  if (baca_adc[iterasi]>batas_hold[iterasi])
  sensor_fikrul[iterasi]=1;
  else
  sensor_fikrul[iterasi]=0;
//    Serial.print(iterasi);
 }
//  Serial.print(baca_adc[1]); Serial.print("\t");
//  Serial.print(baca_adc[2]); Serial.print("\t");
//  Serial.print(baca_adc[3]); Serial.print("\t");
```

```
//  Serial.print(baca_adc[4]); Serial.print("\t");
//  Serial.print(baca_adc[5]); Serial.print("\t");
//  Serial.println(baca_adc[6]); //Serial.print("\r");
//
//  Serial.print(sensor_fikrul[1]); Serial.print("\t");
//  Serial.print(sensor_fikrul[2]); Serial.print("\t");
//  Serial.print(sensor_fikrul[3]); Serial.print("\t");
//  Serial.print(sensor_fikrul[4]); Serial.print("\t");
//  Serial.print(sensor_fikrul[5]); Serial.print("\t");
//  Serial.println(sensor_fikrul[6]);

  sensor_garis =
((sensor_fikrul[6]*32)+(sensor_fikrul[5]*16)+(sensor_fikrul[4]*8)+(sensor_fikrul[3]*4
)+(sensor_fikrul[2]*2)+(sensor_fikrul[1]*1));
  //Serial.println(sensor_garis); Serial.print("\t");

  switch (sensor_garis)
  {
    case 0b110011: PV=0;bipedal.write(95);k=0; break;

    case 0b111011: PV=10 ;bipedal.write(99);k=99-95; break;   //error garis kanan
    case 0b111001: PV=20 ;bipedal.write(103);k=103-95; break;
    case 0b110001: PV=30 ;bipedal.write(107) ;k=107-95;break;
    case 0b111101: PV=40 ;bipedal.write(111);k=111-95;break;
    case 0b111100: PV=50 ;bipedal.write(115);k=115-95;break;
    case 0b111010: PV=60 ;bipedal.write(120);k=120-95;break;
    case 0b111000: PV=70 ; bipedal.write(123);k=123-95;break;
    case 0b110000: PV=80 ;bipedal.write(127);k=127-95;break;
    case 0b100000: PV=90 ;bipedal.write(131);k=131-95;break;
    case 0b111110: PV=100 ;bipedal.write(135);k=135-95;break;


    case 0b110111: PV=-10 ;bipedal.write(91);k=91-95;break;     //error garis kiri
    case 0b100111: PV=-20 ;bipedal.write(87);k=87-95;break;
```

```
    case 0b100011: PV=-30 ;bipedal.write(83);k=83-95;break;
    case 0b101111: PV=-40 ;bipedal.write(79);k=79-95;break;
    case 0b001111: PV=-50 ;bipedal.write(75);k=75-95;break;
    case 0b010111: PV=-60 ;bipedal.write(70);k=70-95;break;
    case 0b000111: PV=-70 ;bipedal.write(67);k=67-95;break;
    case 0b000011: PV=-80 ;bipedal.write(63);k=63-95;break;
    case 0b000001: PV=-90 ;bipedal.write(59);k=59-95;break;
    case 0b011111: PV=-100 ;bipedal.write(55);k=55-95;break;

  }
}
void motor_dc (int duty)
{
  pwm_kiri=180*duty/100;
  pwm_kanan=255*duty/100;
  analogWrite (12, pwm_kiri);
  digitalWrite (10, HIGH);
  digitalWrite (11,LOW);
  analogWrite (13, pwm_kanan);
  digitalWrite (10, HIGH);
  digitalWrite (11,LOW);
}
```

LAMPIRAN IIIDATA SHEET

ARDUINO MEGA 2560



The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4

UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

Technical specs

Microcontroller	**ATmega2560**
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 Ma
DC Current for 3.3V Pin	50 Ma
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Programming

The Mega 2560 board can be programmed with the Arduino Software (IDE). For details, see thereference and tutorials.

The ATmega2560 on the Mega 2560 comes preprogrammed with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in theArduino repository. The

ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

Warnings

The Mega 2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Power

The Mega 2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- Vin. The input voltage to the board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.
- IOREF. This pin on the board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

See the mapping between Arduino pins and Atmega2560 ports:

PIN MAPPING	Atmega2560

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(),digitalWrite(), and digitalRead() functions.

They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50 k ohm. A maximum of 40mA is the value that must not be exceeded to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.
- External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low level, a rising or falling edge, or a change in level. See the attachInterrupt() function for details.
- PWM: 2 to 13 and 44 to 46. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using theSPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Arduino /Genuino Uno and the old Duemilanove and Diecimila Arduino boards.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- TWI: 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library. Note that these pins are not in the same location as the TWI pins on the old Duemilanove or Diecimila Arduino boards.

See also the mapping Arduino Mega 2560 PIN diagram.

The Mega 2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin

and analogReference() function.

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Mega 2560 board has a number of facilities for communicating with a computer, another board, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega16U2 (ATmega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2/ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega 2560's digital pins.

The Mega 2560 also supports TWI and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the TWI bus; see the documentation for details. For SPI communication, use the SPI library.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega 2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the

board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega 2560 is designed to be compatible with most shields designed for the Uno and the older Diecimila or Duemilanove Arduino boards. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Furthermore, the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega 2560 and Duemilanove / Diecimila boards. Please note that I2C is not located on the same pins on the Mega 2560 board (20 and 21) as the Duemilanove / Diecimila boards (analog inputs 4 and 5).

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Mega 2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega 2560 board is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the ATMega2560. While it is programmed to ignore malformed data (i.e.

anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega 2560 board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

Revisions

The Mega 2560 does not use the FTDI USB-to-serial driver chip used in past designs. Instead, it features the ATmega16U2 (ATmega8U2 in the revision 1 and revision 2 Arduino boards) programmed as a USB-to-serial converter. Revision 2 of the Mega 2560 board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the Arduino board and the current Genuino Mega 2560 have the following improved features:

- 1.0 pinout: SDA and SCL pins - near to the AREF pin - and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the board that uses ATSAM3X8E, that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2

Batterai Lippo 12V

NO.	Spesification	Detail
1.	Minimum Capacity	**2200mAh (True 100% Capacity)**
2.	Configuration	**3S1P / 11.1v / 3Cell**
3.	Constant Discharge	**20C**
4.	Peak Discharge (10sec)	**30C**
5.	Pack Weight	**188g**
6.	Pack Size	**103 x 33 x 24mm**
7.	Charge Plug	**JST-XH**
8.	Discharge Plug	**XT60**