

KATA PENGANTAR

Assalamualaikum wr.wb. Alhamdulillah, puji dan syukur penulis panjatkan kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk mencapai gelar Sarjana Teknik dari jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang.

Penulis menyadari bahwa tanpa bantuan, bimbingan serta dorongan dari semua pihak penyelesaian skripsi ini tidak mungkin bisa terwujud. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar – besarnya kepada :

1. Allah SWT yang telah memberikan kelancaran, kemudahan dan hidayahNya.
2. Ayah dan Ibu tercinta serta adik kakak yang telah banyak memberikan kasih sayang, dukungan, doa serta semangat.
3. Istri saya tercinta Fitri Indah Sasmitha yang selalu setia mendampingi, memotivasi dan mendoakan dalam pengerjaan skripsi ini.
4. Bapak M. Aziz Muslim, ST., MT., Ph.D. selaku ketua jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya Malang.
5. Bapak Hadi Suyono ST., MT., Ph.D selaku Sekretaris jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.
6. Bapak Ir. Purwanto, MT. selaku KKDK Sistem Kontrol.
7. Rahmadwati ST., MT., Phd selaku dosen pembimbing skripsi yang telah banyak memberikan pengarahan, bimbingan dalam penyelesaian ini serta atas segala bentuk bantuan, saran dan motivasinya yang membangun.
8. Bambang Siswoyo ST., MT selaku dosen pembimbing skripsi yang telah banyak memberikan pengarahan, bimbingan dalam penyelesaian ini serta atas segala bentuk bantuan, saran dan motivasinya yang membangun.

9. Bapak, Ibu dosen serta segenap staf dan karyawan Jurusan Teknik Elektro baik secara langsung maupun tidak langsung yang telah banyak membantu dalam menyelesaikan skripsi ini.
10. Rekan – rekan Asisten Laboratorium Sistem Kontrol.
11. Rekan – rekan mahasiswa Teknik Elektro Brawijaya, keluarga Ampere '09 serta sahabat – sahabatku terima kasih untuk semuanya.
12. Imam, Yudi, Deaz, dan Bambang terima kasih telah menjadi tim terbaik sepanjang masa.
13. Praja, Zila, Bosje, Gopar, Semigol, Rido, Rokim, Ade, Cheny, Arga, Pepi, Iksan, Suryo Terri, Rafi, Pakde, Sopir, Temmy, Andik, Taufiq, Rio, Ari yang telah sudi menjadi sahabat yang baik dan selalu memotivasi.
14. Semua pihak yang telah memberikan bantuan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini belumlah sempurna, karena keterbatasan ilmu dan kendala – kendala lain yang terjadi selama pengerjaan skripsi ini. Semoga tulisan ini dapat bermanfaat dan dapat digunakan untuk pengembangan lebih lanjut. Wassalamualaikum wr.wb.

Malang, Maret 2014

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	vi
DAFTAR TABEL	viii
ABSTRAK	ix
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Sistematika Penulisan.....	3
TINJAUAN PUSTAKA	4
2.1 Robot <i>Line Follower</i>	4
2.2 Sensor Cahaya.....	4
2.3 Motor DC.....	5
2.4 Kontroler.....	7
2.4.1 Kontroler Proporsional.....	7
2.4.2 Kontroler Integral	8
2.4.3 Kontroler Diferensial	9
2.4.4 Kontroler Proporsional Integral Diferensial (PID).....	10
2.4.5 Metode Perancangan Kontroler Proporsional Integral Diferensial (PID) Menggunakan Metode Ziegler-Nichols.....	12
2.5 <i>Pulse Width Modulation (PWM)</i>	15
2.6 Mikrokontroler	16
2.6.1 ATMEGA8.....	16
2.6.2 ATMEGA162.....	18
2.7 Program CodeVision AVR.....	22
2.8 Rotary Encoder.....	24
METODOLOGI	26
3.1 Studi literatur.....	26

3.2	Perancangan Blok Diagram Sistem.....	26
3.3	Prinsip Kerja	26
3.4	Perancangan Perangkat Keras	27
3.5	Perancangan Perangkat Lunak.....	27
3.6	Pengujian dan Analisis Sistem.....	27
3.7	Pengambilan Kesimpulan dan Saran.....	27
	PERANCANGAN DAN PEMBUATAN ALAT	28
4.1	Spesifikasi Alat	28
4.2	Diagram Blok Sistem	28
4.3	Perancangan Perangkat Keras.....	29
4.3.1	Perancangan Rangka Robot <i>Line Follower Berbeban</i>	29
4.3.2	Rangkaian <i>Board</i> Mikrokontroler	30
4.3.3	Rotary Encoder.....	32
4.3.4	Pemilihan Motor DC.....	32
4.4	Perancangan Sistem Tuning Kontroler PID Menggunakan Metode Kedua Ziegler-Nichols.....	33
4.4.1	Perancangan <i>Tuning</i> Parameter PID Pada Motor 1 (Kanan)	33
	Gambar 4.6 Grafik respon kecepatan roda 1 dengan $K_p=5$, $K_i=0$, $K_d=0$ ($K_{cr} = 5$)	33
4.4.2	Perancangan <i>Tuning</i> Parameter PID Pada Motor 2 (Kiri)	34
4.4	Perancangan Perangkat Lunak.....	35
	PENGUJIAN DAN ANALISIS SISTEM.....	33
5.1	Pengujian Sensor Rotary Encoder.....	33
5.1.1	Pengujian Keluaran Sensor Rotary Encoder.....	33
5.1.2	Pengujian Rotary Encoder Sebagai Sensor Kecepatan.....	34
5.2	Pengujian Hasil <i>Tuning</i> Parameter PID.....	35
5.3	Pengujian Keseluruhan Sistem.....	36
5.3.1	Peralatan Pengujian	37
5.3.2	Prosedur Pengujian	37
5.3.3	Hasil Pengujian.....	37
	PENUTUP	43
6.1	Kesimpulan	43

6.2 Saran..... 43
DAFTAR PUSTAKA..... 44



DAFTAR GAMBAR

Gambar 2.1 Hubungan keluaran *photodiode* dengan intensitas cahaya..... 4

Gambar 2.2 Ilustrasi mekanisme sensor garis..... 5

Gambar 2.3 Ilustrasi Cara Kerja Motor DC 6

Gambar 2.4 Diagram Blok Kontroler Proporsional (Ogata K., 1997)..... 8

Gambar 2.5 Diagram Blok Kontroler Integral (Ogata K., 1997)..... 9

Gambar 2.6 Diagram Blok Kontroler Diferensial (Ogata K., 1997) 10

Gambar 2.7 Diagram Blok Kontroler PID (Ogata K., 1997) 11

Gambar 2.8 Kurva Respon Unit Step Menunjukkan 25% *Maximum Overshoot* (Ogata K., 1997)..... 12

Gambar 2.9 Respon Plant Terhadap Masukan Berupa Unit Step (Ogata K., 1997) 12

Gambar 2.10 Kurva yang Berbentuk S (Ogata K., 1997) 13

Gambar 2.11 Sistem Loop Tertutup dengan Kontroler Proporsional (Ogata K., 1997) 14

Gambar 2.12 Osilasi Berkesinambungan dengan Periode Pcr (Ogata K., 1997) 14

Gambar 2.13 Sinyal PWM Secara Umum (insansains.wordpress.com, 2008) 15

Gambar 2.14 Konfigurasi Pin ATmega8..... 17

Gambar 2.15 . Pin-Pin pada ATmega162 dengan Kemasan 40-Pin DIP (*Dual In-Line Package*) 20

Gambar 2.16 Rotary Encoder 24

Gambar 2.17 *Line Driver Output* Sensor Rotary Encoder..... 25

Gambar 3.1 Blok Diagram Sistem..... 26

Gambar 4.1 Blok Diagram Sistem..... 29

Gambar 4.2 Rangka Robot *Line Follower* Berbeban (Perancangan, 2013)... 30

Gambar 4.3 Rangkaian Minimum Sistem Mikrokontroler Master..... 30

Gambar 4.4 Rangkaian Minimum Sistem Mirokontroler *Slave* Motor..... 31

Gambar 4.5 Rotary Encoder (Sumber: Perancangan)..... 32

Gambar 4.5 Motor DC dan Rangkaian *Gear box* 32

Gambar 4.6 Grafik respon kecepatan roda 1 dengan $K_p=5, K_i=0, K_d=0$ ($K_{cr} = 5$)..... 33

Gambar 4.7 Grafik respon kecepatan motor 2 dengan $K_p=4,9, K_i=0, K_d=0$ ($K_{cr} = 4,9$)..... 34

Gambar 4.8 *Flowchart* Perangkat Lunak (Perancangan, 2013)..... 35

Gambar 5.1 Diagram Blok Pengujian Keluaran Sensor Rotary Encoder..... 33

Gambar 5.2 Diagram Blok Pengujian Rotary Encoder Sebagai Sensor Kecepatan..... 34

Gambar 5.3 Diagram Blok Pengujian Hasil *Tuning* Parameter Kontroler PID..... 36

Gambar 5.4 Grafik Respon Kecepatan Motor 1..... 36

Gambar 5.5 Grafik Respon Kecepatan Motor 2..... 36

Gambar 5.6 Respon Motor 1 dengan beban box menggunakan kontrol PID 37

Gambar 5.7 Respon Motor 2 dengan beban box menggunakan kontrol PID 38

Gambar 5.8 Respon Motor 1 dengan box dan beban 2,5 kg menggunakan kontrol PID 38

Gambar 5.9 Respon Motor 2 dengan box dan beban 2,5 kg menggunakan kontrol PID 38

Gambar 5.10 Respon Motor 1 dengan box dan beban 5 kg menggunakan kontrol PID 39

Gambar 5.11 Respon Motor 2 dengan box dan beban 5 kg menggunakan kontrol PID 39

Gambar 5.12 Respon Motor 1 dengan box dan beban 7,5 kg menggunakan kontrol PID 39

Gambar 5.13 Respon Motor 2 dengan box dan beban 7,5 kg menggunakan kontrol PID 40

DAFTAR TABEL

Tabel 2.1 Aturan Penalaran Ziegler-Nichols Berdasarkan Respon Unit Step dari Plan (Ogata K., 1997)..... 13

Tabel 2.2 Aturan Dasar Ziegler-Nichols Berdasarkan *Critical Gain* Ker dan *Critical Period* Pcr (Ogata K., 1997)..... 15

Tabel 2.3 Konfigurasi Pin *Rotary Encoder*..... 24

Tabel 5.1 Hasil Pengujian data pulsa *rotary encoder*..... 34

Tabel 5.2 Hasil Pengujian Kecepatan motor dengan Rotary Encoder Dan Tachometer 35

Tabel 5.3 Tabel Data Performansi Sistem..... 40



ABSTRAK

Eka Bayu Prinandika, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Maret 2014, *Sistem Pengaturan Kecepatan Motor Pada Robot Line Follower Berbeban Menggunakan Kontroler PID*, Dosen Pembimbing: Rahmadwati ST.,MT.,Phd. dan Bambang Siswoyo ST., MT.

Abstrak- . Laporan ini membahas tentang system pengaturan kecepatan motor pada robot line follower berbeban. Digunakan Kontroler PID agar kecepatan motor dapat bertahan pada nilai kecepatan yang diinginkan meskipun dengan beban robot yang berubah-ubah. Pada skripsi ini digunakan metode *Ziegler Nichols* Dalam pembuatannya digunakan mikrokontroler ATmega8, sensor *rotary encoder* dan motor DC.

Dari hasil pengujian terhadap aplikasi kontroler PID dengan menggunakan metode *Ziegler Nichols*. Didapatkan parameter PID dengan nilai $K_p=3, K_i= 10, K_d=0.225$ untuk motor 1 dan $K_p=4.9, K_i=9.8 K_d=0,2205$ untuk motor 2. Rata-rata *error steady state* dari pengujian yang telah dilakukan adalah sebesar 1,19%.

Kata Kunci : *Rotary encoder, mikrokontroler ATmega8, Motor DC, PID.*

BAB I PENDAHULUAN

1.1 Latar Belakang

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

Robot memiliki berbagai macam konstruksi, antara lain: robot mobil (*mobile robot*) yang dapat berpindah posisi dari satu titik ke titik yang lain, robot manipulator (tangan) yang dapat bekerja menyerupai lengan manusia, robot *humanoid* yang menyerupai manusia secara utuh, *flying robot* yang dapat terbang, robot berkaki, robot jaringan yang dikendalikan melalui internet dengan protocol TCP/IP, robot animalia, dan robot *cybrog*.

Dari macam konstruksi yang ada, *mobile robot* sangat disukai bagi orang yang mulai mempelajari robot. Karena tidak memerlukan kerja fisik yang berat dan minimal hanya diperlukan pengetahuan tentang mikrokontroler dan sensor-sensor elektronik. *Mobile robot* dapat dibuat sebagai pengikut garis (*Line Follower*) atau pengikut dinding (*Wall Follower*) ataupun pengikut cahaya.

Dari ketiganya yang paling umum adalah robot *Line Follower*. *Line Follower* merupakan jenis robot beroda yang bergerak berdasarkan jalur garis (*Line*) yang sudah ditentukan dengan daya menggerak berupa motor. Namun yang sering terjadi adalah robot bergerak tidak stabil dalam mengikuti jalur, misalnya pada saat *track* lurus posisi sudut badan robot tidak nol derajat (0°) dengan *Line*, akibatnya robot bergerak melenceng, berbelok-belok, atau bisa juga robot bergerak dengan sudut badan robot nol derajat (0°) dengan *Line* namun posisi robot tidak simetris dengan *Line*.

Laporan ini membahas tentang sistem pengaturan kesetabilan kecepatan motor pada robot *line follower* berbeban. Kontroler yang digunakan dalam laporan ini adalah Kontroler Proporsional Integral Differential (PID) berbasis mikrokontroler ATmega8 dan ATmega16. PID adalah kontroler yang merupakan gabungan dari kontroler proporsional, kontroler integral dan kontroler differensial. Gabungan dari ketiga kontroler ini diharapkan agar mendapat keluaran sistem yang stabil karena bisa saling menutupi kekurangan. Keuntungan dari kontroler PID adalah merupakan sebuah sistem yang sederhana sehingga lebih cepat dalam mengambil sebuah keputusan. Diharapkan dengan menggunakan kontroler PID kesetabilan kecepatan motor terjaga pada nilai yang diinginkan, dan reaksi sistem yang didapatkan cepat. Sehingga hasilnya dapat dijadikan referensi atau kontribusi ilmiah untuk membangun sebuah robot yang berguna bagi kehidupan manusia.

1.2 Rumusan Masalah

Dari latar belakang masalah di atas, dibuat rumusan masalah sebagai berikut :

1. Bagaimana merancang *hardware* sistem pengaturan kecepatan motor pada robot *line follower berbeban* menggunakan kontroler PID?
2. Bagaimana merancang *software* sistem pengaturan kecepatan motor pada robot *line follower berbeban* menggunakan kontroler PID?

1.3 Batasan Masalah

Dalam perancangan, masalah-masalah yang dibatasi adalah sebagai berikut :

1. Pembahasan ditekankan pada penggunaan kontroler PID pada sistem
2. Sensor yang digunakan adalah rotary encoder.
3. Aktuator yang digunakan adalah motor DC dengan kecepatan 1930 rpm.
4. Pemrograman mikrokontroler (*slave*) menggunakan software *CodeVisionAVR Compiler* untuk pengolahan data
5. Mikrokontroler yang digunakan yaitu 3 buah Atmega 8 dan 1 buah mikrokontroler master Atmega 16.
6. Dimensi robot yaitu panjang 100 cm, lebar 40 cm dan tinggi 45 cm.
7. Perhitungan fisika yang terjadi pada sistem tidak dibahas mendalam.
8. Kinerja *driver* dan elektronika tidak dibahas mendalam.

1.4 Tujuan

Merancang sistem kendali yang dapat mengatur kecepatan motor pada robot dengan beban yang berubah-ubah menggunakan kontroler PID.

1.5 Sistematika Penulisan

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

BAB I Pendahuluan

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

BAB II Tinjauan Pustaka

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

BAB III Metodologi Penelitian

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

BAB IV Perancangan dan Pembuatan Alat

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

BAB V Pengujian dan Analisis

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

BAB VI Penutup

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

BAB II

TINJAUAN PUSTAKA

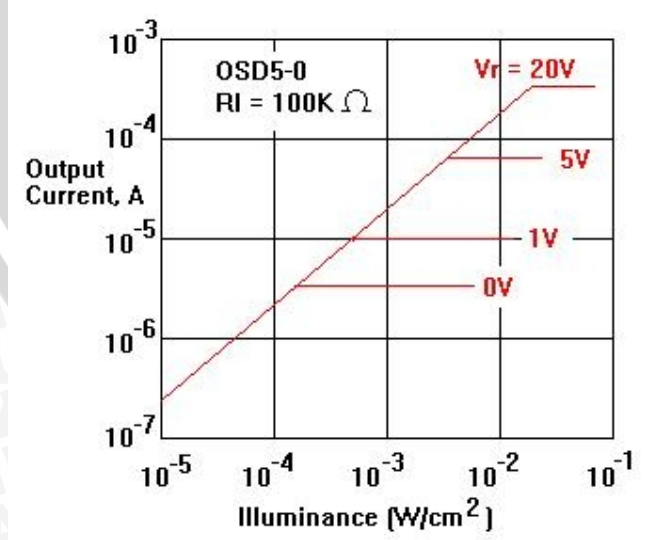
2.1 Robot Line Follower

Robot *line follower* adalah robot yang bergerak secara otomatis mengikuti lika-liku garis di permukaan lantai. Untuk mengenali pola garis, digunakan sensor cahaya yang akan mendeteksi terang gelap permukaan yang ada di bawahnya. Untuk itu, harus diperoleh perbedaan warna yang mencolok antara warna garis dan warna permukaan lantai.

2.2 Sensor Cahaya

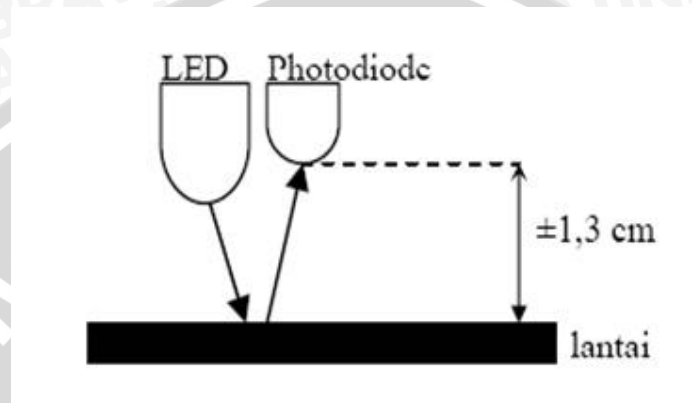
Sensor *photodiode* adalah salah satu jenis sensor yang peka terhadap cahaya (*photodetector*). *Photodiode* akan mengalirkan arus yang membentuk fungsi linear terhadap intensitas cahaya yang diterima. Arus ini umumnya teratur terhadap *power density* (D_p). Perbandingan antara arus keluaran dengan *power density* disebut sebagai *current responsivity*. Arus yang dimaksud adalah arus bocor ketika *photodiode* tersebut disinari dan dalam keadaan dipanjar mundur.

Hubungan antara keluaran sensor *photodiode* dengan intensitas cahaya yang diterimanya ketika dipanjar mundur adalah membentuk suatu fungsi yang linear. Gambar 2.1 menunjukkan grafik hubungan antara keluaran sensor *photodiode* dengan intensitas cahaya.



Gambar 2.1 Hubungan keluaran *photodiode* dengan intensitas cahaya
 (Sumber: www.fahmizaleeits.wordpress.com)

LED *superbright* berfungsi sebagai pengirim cahaya ke garis untuk dipantulkan lalu dibaca oleh sensor *photodiode*. Sifat dari warna putih (permukaan terang) yang memantulkan cahaya dan warna hitam (permukaan gelap) yang tidak memantulkan cahaya digunakan dalam aplikasi ini. Gambar dibawah ini adalah ilustrasi mekanisme sensor garis.



Gambar 2.2 Ilustrasi mekanisme sensor garis
(Sumber: www.fahmizaleeits.wordpress.com)

2.3 Motor DC

Prinsip kerja motor DC sesuai dengan hukum kemagnetan Lorentz, yaitu membangkitkan fungsi magnet pada suatu konduktor berarus dalam medan magnet sehingga timbul ggl induksi. Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kiri.

Kaidah tangan kiri untuk motor menunjukkan arah arus yang mengalir didalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah medan putar. Adapun besarnya gaya yang bekerja pada konduktor tersebut dapat dirumuskan dengan:

$$F=B.I.L \quad (\text{Newton}) \quad (2.1)$$

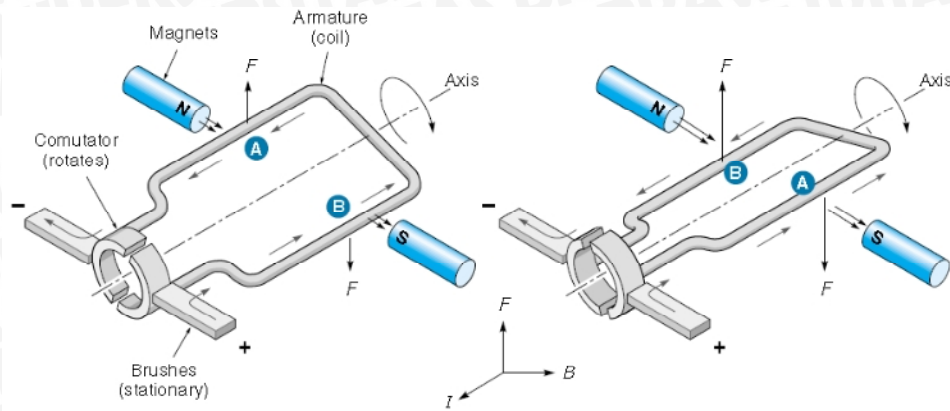
Dimana :

B = kerapatan fluks magnet (weber)

L = panjang konduktor (meter)

I = arus listrik (ampere)

(Soemarwanto,1999)



Gambar 2.3 Ilustrasi Cara Kerja Motor DC
(Sumber: Killian, 1996)

Gambar 2.3 adalah ilustrasi cara kerja motor DC yang mempunyai satu lilit kawat a-b berada di dalam medan magnet. Lilitan ini dapat berputar dengan bebas, lilitan ini biasa disebut dengan jangkar (*armature*).

Pada jangkar diberikan arus yang berasal dari sumber yang terhubung dengan sikat (*brushes*). Sikat-sikat ini terpasang pada sebuah cincin yang terbelah dua, yang disebut cincin belah (*commutator*). Adapun tujuan dari konstruksi ini adalah agar lilitan kawat dapat berputar apabila ada arus listrik yang melewatinya.

Pada kawat yang berada di kanan arus mengalir dari depan ke belakang. Pada kawat yang berada di bagian kiri, arus mengalir dari belakang ke depan kawat a dan b secara bergantian berada di kiri dan kanan. Karena itu arah arus di a dan arah arus di b selalu bersifat bolak-balik. Pembalikan arah arus itu terjadi pada saat lilitan kawat melintasi posisi vertical.

Bagian *commutator* berfungsi sebagai penyearah mekanik. Fluksi magnet yang ditimbulkan magnet permanen disebut medan magnet motor. Dalam gambar 2.3 arah fluks magnetic adalah dari kiri ke kanan. Adapun gaya yang bekerja pada penghantar b adalah ke atas, sementara gaya yang bekerja pada penghantar a adalah ke bawah. Gaya-gaya yang bekerjasama kuatnya, sehingga terdapat kopel yang bekerja pada kawat sehingga lilitan jangkar dapat berputar. Setelah berputar 180° arah arus berbalik, pada saat itu penghantar a dan b bertukar tempat. Akibatnya arah gerak putaran tidak berubah.

2.4 Kontroler

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

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

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

2.4.1 Kontroler Proporsional

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

$$m(t) = K_p e(t) \dots\dots\dots (2.2)$$

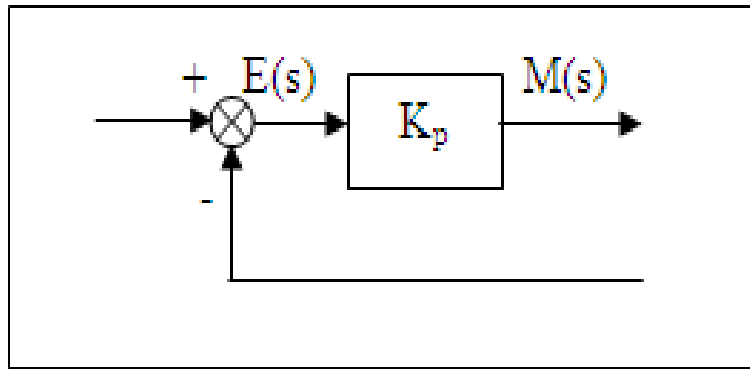
atau, dalam besaran transformasi Laplace,

$$\frac{M(s)}{E(s)} = K_p \dots\dots\dots (2.3)$$

di mana K_p adalah kepekaan proporsional atau penguatan.

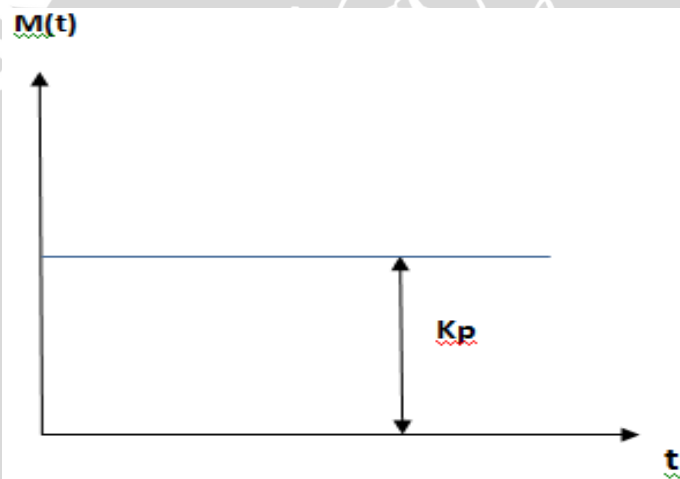
Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggeraknya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K.,1997).

Diagram blok kontroler proporsional ditunjukkan dalam Gambar 2.4.



Gambar 2.4 Diagram Blok Kontroler Proporsional (Ogata K., 1997)

Gambar Output Kontroler proporsional jika diberi input step ditunjukkan pada gambar 2.5.



Gambar 2.5 Grafik Respon Kontroler Proporsional Untuk Input Unit Step

2.4.2 Kontroler Integral

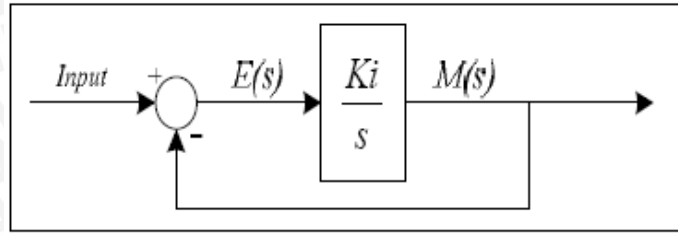
Pada kontroler dengan aksi integral, harga keluaran kontroler $m(t)$ diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak $e(t)$.

Jadi,

$$\frac{dm(t)}{dt} = Kie(t) \dots\dots\dots (2.4)$$

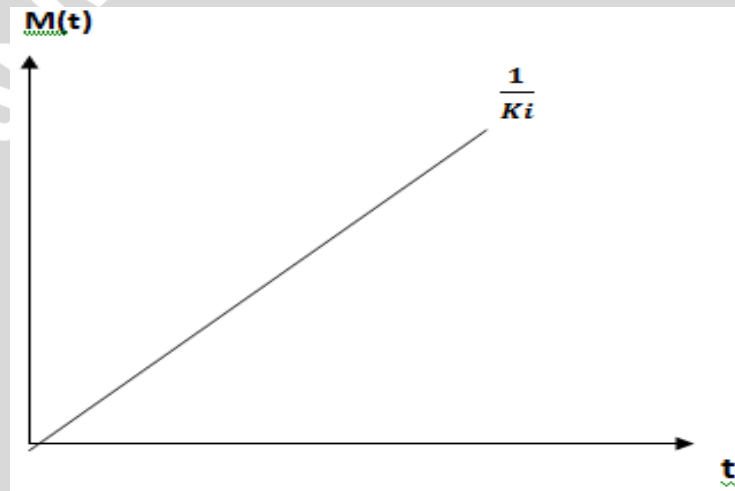
dengan K_i adalah konstanta integral. Jika harga $e(t)$ diduakalikan, maka harga $m(t)$ berubah dengan laju perubahan menjadi dua kali semula. Jika kesalahan penggerak nol, maka harga $m(t)$ tetap stasioner. Aksi kontrol integral seringkali disebut kontrol *reset* yang digunakan untuk menghilangkan *error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

Diagram blok kontroler integral ditunjukkan dalam Gambar 2.5.



Gambar 2.6 Diagram Blok Kontroler Integral (Ogata K., 1997)

Gambar Output Kontroler integral jika diberi input step ditunjukkan pada gambar 2.7.



Gambar 2.7 Grafik Respon Kontroler Integral Untuk Input Unit Step

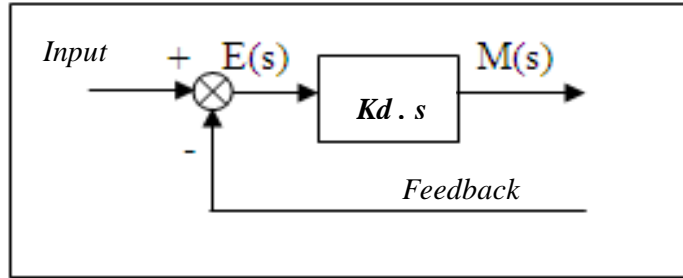
2.4.3 Kontroler Diferensial

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

$$\frac{M(s)}{E(s)} = Kd \cdot s \dots\dots\dots (2.5)$$

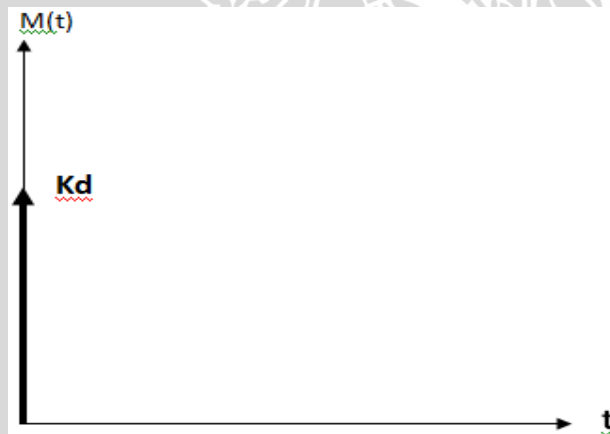
dengan Kd adalah konstanta diferensial yang digunakan untuk memperbaiki atau mempercepat respon transien sebuah sistem serta dapat meredam osilasi.

Gambar 2.6 menunjukkan diagram blok kontroler diferensial.



Gambar 2.8 Diagram Blok Kontroler Diferensial (Ogata K., 1997)

Gambar Output Kontroler diferensial jika diberi input step ditunjukkan pada gambar 2.9.



Gambar 2.9 Grafik Respon Kontroler Diferensial Untuk Input Unit Step

2.4.4 Kontroler Proporsional Integral Diferensial (PID)

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

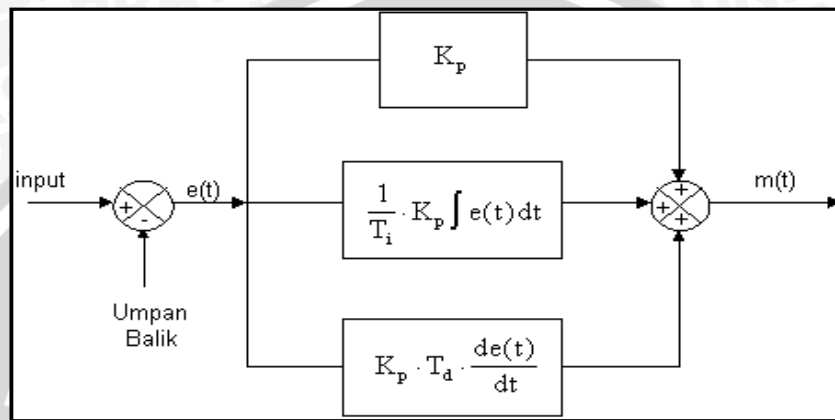
$$m(t) = Kp \cdot e(t) + \frac{Kp}{Ti} \cdot e(t)dt + Kp \cdot Td \frac{de(t)}{dt} \dots\dots\dots(2.6)$$

Dalam transformasi Laplace dinyatakan sebagai berikut :

$$\frac{M(s)}{E(s)} = Kp \left(1 + \frac{1}{Ti \cdot s} + Td \cdot s \right) \dots\dots\dots(2.7)$$

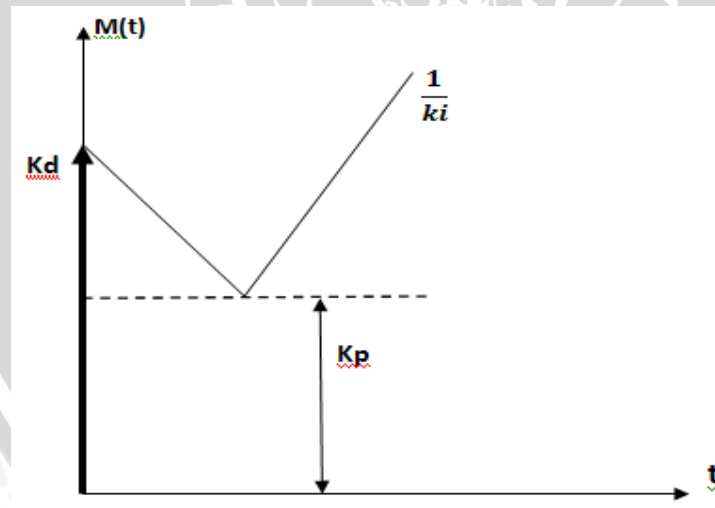
dengan Ti adalah waktu integral, dan Td adalah waktu derivatif.

Gambar 2.7 menunjukkan diagram blok kontroler PID.



Gambar 2.10 Diagram Blok Kontroler PID (Ogata K., 1997)

Gambar Output Kontroler proporsional integral diferensial jika diberi input step ditunjukkan pada gambar 2.8.



Gambar 2.11 Grafik Respon Kontroler Proporsional Integral Diferensial Untuk Input Unit Step

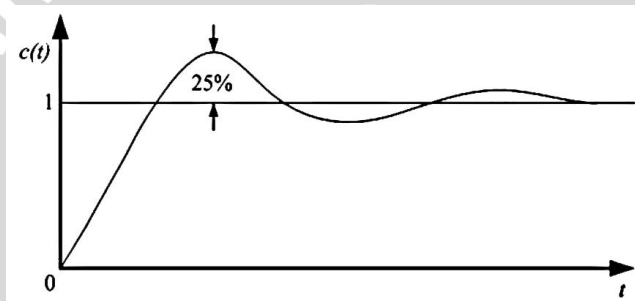
2.4.5 Metode Perancangan Kontroler Proporsional Integral Diferensial (PID)

Menggunakan Metode Ziegler-Nichols

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

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

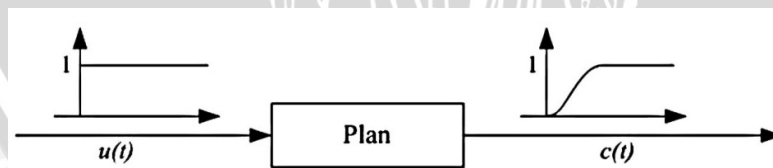
Kurva respon unit step yang menunjukkan 25% *maximum overshoot* terlihat dalam Gambar 2.12.



Gambar 2.12 Kurva Respon Unit Step Menunjukkan 25% *Maximum Overshoot* (Ogata K., 1997)

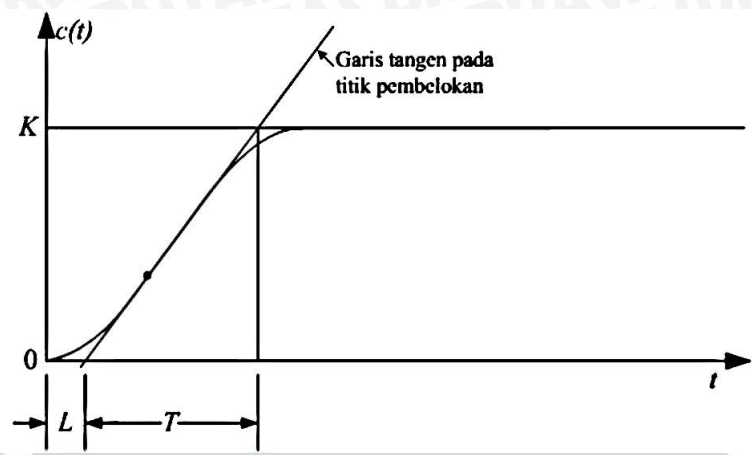
a) Metode Pertama

Metode pertama atau sering disebut metode kurva reaksi, respon dari plan dapat dapat diperoleh secara eksperimental dengan masukan berupa unit *step*, seperti yang ditunjukkan dalam Gambar 2.13.



Gambar 2.13 Respon Plant Terhadap Masukan Berupa Unit Step (Ogata K., 1997)

Jika dalam plan tersebut terdapat integrator atau *dominan complex-conjugate poles*, maka kurva respon unit step berbentuk seperti huruf S, seperti dalam Gambar 2.14, jika respon tidak memberikan bentuk kurva S, maka metode ini tidak berlaku. (Ogata K., 1997).



Gambar 2.14 Kurva yang Berbentuk S (Ogata K., 1997)

Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda L dan konstanta waktu T . Waktu tunda dan konstanta waktu ditentukan dengan menggambar sebuah garis tangen pada titik pembelokan dari kurva S, dan menentukan perpotongan antara garis tangen dengan sumbu waktu t dan sumbu $c(t) = K$, seperti yang telah ditunjukkan dalam Gambar 2.10. Fungsi alih $C(s)/U(s)$ dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut:

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts+1} \dots\dots\dots(2.8)$$

Ziegler dan Nichols menyarankan untuk menentukan nilai-nilai dari K_p , T_i dan T_d berdasarkan pada formula yang ditunjukkan dalam Tabel 2.1 (Ogata K., 1997).

Tabel 2.1 Aturan Penalaran Ziegler-Nichols Berdasarkan Respon Unit Step dari Plan (Ogata K., 1997)

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

Aturan untuk metode pertama dengan persamaan sebagai berikut:

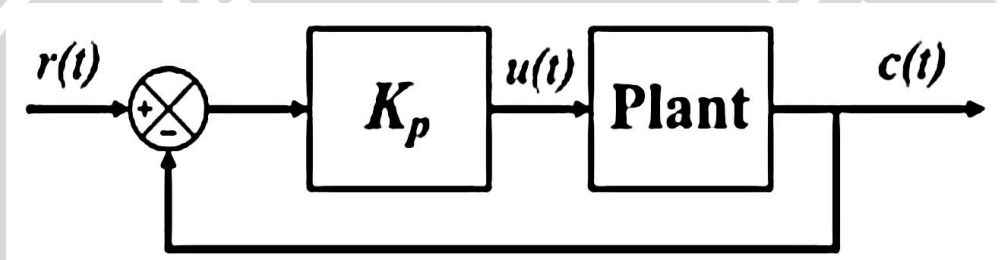
$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = 1.2 \frac{T}{L} \left(1 + \frac{1}{2Ls} + 0.5Ls \right) \dots\dots\dots(2.9)$$

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

b) Metode Kedua

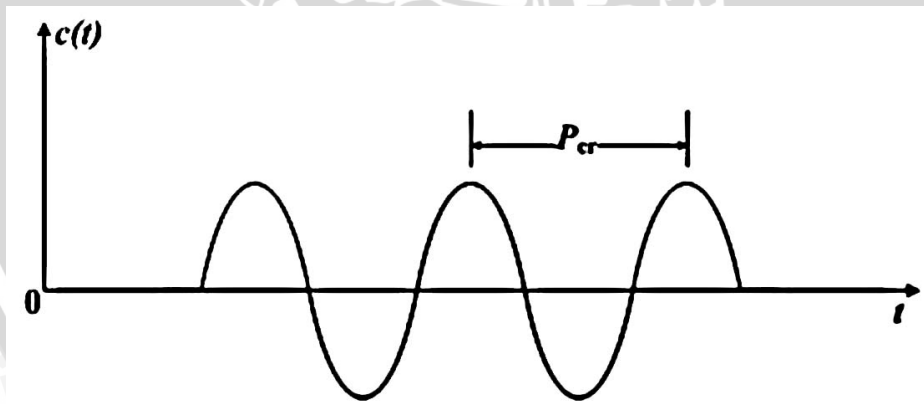
Dalam metode kedua Ziegler-Nichols, mula-mula yang dilakukan adalah membuat $T_i = \infty$ dan $T_d = 0$. Kemudian hanya dengan menggunakan tindakan kontrol proporsional, harga ditingkatkan dari nol ke suatu nilai kritis K_{cr} , disini mula-mula keluaran memiliki osilasi yang berkesinambungan (Jika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

Diagram blok sistem *loop* tertutup dengan kontroler proporsional dapat dilihat dalam Gambar 2.15.



Gambar 2.15 Sistem Loop Tertutup dengan Kontroler Proporsional (Ogata K., 1997)

Osilasi berkesinambungan dengan periode P_{cr} dapat dilihat dalam Gambar 2.16.



Gambar 2.16 Osilasi Berkesinambungan dengan Periode P_{cr} (Ogata K., 1997)

Ziegler dan Nichols menyarankan penyetelan nilai parameter K_p , T_i , dan T_d berdasarkan rumus yang diperlihatkan dalam Tabel 2.2 (Ogata K., 1997).

Tabel 2.2 Aturan Dasar Ziegler-Nichols Berdasarkan *Critical Gain* Kcr dan *Critical Period* Pcr (Ogata K., 1997)

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

Aturan untuk metode kedua dengan persamaan sebagai berikut:

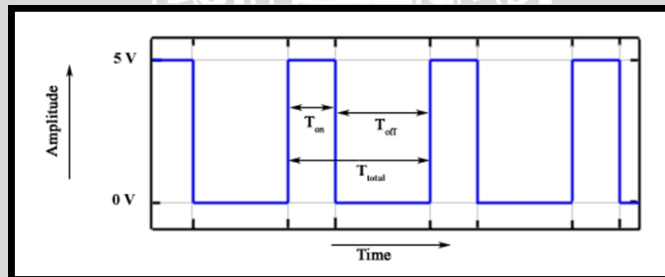
$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = 0.6 K_{cr} \left(1 + \frac{1}{0.5 P_{cr} s} + 0.125 P_{cr} s \right) \dots\dots\dots (2.10)$$

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

2.5 Pulse Width Modulation (PWM)

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

Sinyal *Pulse Width Modulation* (PWM) secara umum dapat dilihat dalam Gambar 2.17 berikut:



Gambar 2.17 Sinyal PWM Secara Umum (insansains.wordpress.com, 2008)

$$Duty\ cycle = \frac{T_{on}}{T} \times 100\% (\%) \dots\dots\dots (2.11)$$

Dengan:

Ton = Periode logika tinggi

T = Periode keseluruhan

$$V_{dc} = \text{Duty cycle} \times V_{cc} \text{ (V)} \dots\dots\dots (2.12)$$

Sedangkan frekuensi sinyal dapat ditentukan dengan rumus berikut:

$$f_{On} = \frac{f_{clk I/O}}{N.256} \text{ (Hz)} \dots\dots\dots (2.13)$$

2.6 Mikrokontroler

2.6.1 ATMEGA8

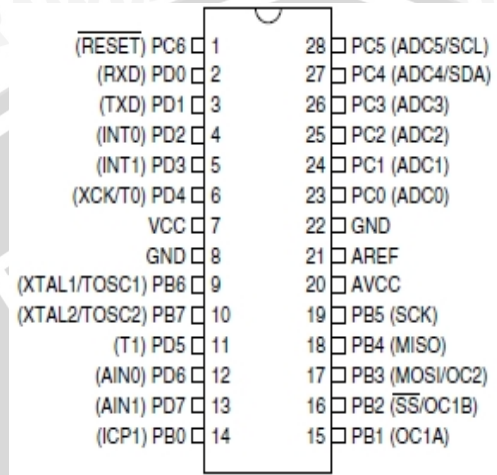
Secara umum, Mikrokontroler adalah suatu IC (*integrated circuit*) yang memiliki kemampuan untuk membuat keputusan berdasarkan sinyal dari luar (input), dan berdasarkan algoritma yang dibentuk menjadi suatu program tertentu. Pada dasarnya mikrokontroler memiliki mikroprosesor, *timer*, *counter*, perangkat I/O dan internal memori.

Sedangkan **ATmega8** menurut datasheet ATmega8 adalah mikrokontroler CMOS 8-bit berarsitektur AVR RISC yang memiliki 8K Bytes *ISP Flash*. Dengan mengeksekusi instruksi dalam *clock* yang sangat cepat, ATmega8 memiliki *throughput* mendekati 1 MIPS per MHZ, yang memungkinkan perancang sistem mengoptimalkan konsumsi daya terhadap kecepatan pemrosesan.

Sebagai suatu sistem kontrol mikrokontroler ATmega8 bila dibandingkan dengan mikroprosesor memiliki kemampuan dan segi ekonomis yang bisa diandalkan karena dalam mikrokontroler sudah terdapat RAM dan ROM sedangkan mikroprosesor didalamnya tidak terdapat keduanya. Secara umum konfigurasi yang dimiliki mikrokontroler ATmega8 adalah sebagai berikut :

- Sebuah CPU 8 bit dengan menggunakan teknologi dari Atmel.
- Jalur dua arah (*bidirectional*) yang digunakan sebagai saluran masukan atau keluaran yang dikontrol oleh *register DDR*.
- Dua buah *timer/counter* 8 bit dan sebuah *timer/counter* 16 bit.
- 23 jalur *programmable I/O* (input/output).
- *Analog to Digital Converter (ADC)* 10-bit dan *Analog comparator* di dalam chip.
- Osilator internal dan rangkaian pewaktu.
- Sebuah komunikasi serial USART yang dapat diprogram.
- Sebuah *master/slave* serial SPI yang dapat diprogram.
- Sebuah *Two Wire Serial Interface*.

Masing-masing pin mikrokontroler ATmega8 mempunyai fungsi tersendiri. Dengan mengetahui fungsi masing-masing pin mikrokontroler ATmega8, perancangan aplikasi mikrokontroler ATmega8 akan lebih mudah dan maksimal. ATmega8 mempunyai 28 pin, susunan masing-masing pin ditunjukkan dalam Gambar 2.18.



Gambar 2.18 Konfigurasi Pin ATmega8
Sumber: ATMEL, 2010

Fungsi kaki-kaki PIN dalam ATmega8 sesuai dengan *data sheet* ATmega8 antara lain sebagai berikut:

- *Port B* (Pin B0..7), merupakan saluran masukan/keluaran dua arah dan juga mempunyai fungsi khusus. Fungsi khusus *Port B* diantaranya adalah : *Port B0* {ICP (*Timer/counter input capture pin*)}, *Port B1* {OC1A (*Timer/counter 1 output compare A match output*)}, *Port B2* yang bisa digunakan sebagai SS (*SPI slave select input*) atau OC1B (*Timer/counter 1 output compare B match output*) , *Port B3* yang bisa digunakan sebagai MOSI (*SPI bus master output/slave input*) atau OC2 (*timer/counter 2 compare match output*), *Port B4* {MISO (*SPI bus master input/slave output*)}, *Port B7* {SCK (*SPI bus serial clock*)}, *Port B5* {SCK (*SPI bus serial clock*)}, *Port B6* (XTAL1 & TOSC1 (*Timer Oscilator pin1*)), dan *Port B7* (XTAL2 & TOSC2 (*Timer Oscilator pin2*)).
- *Port C* (Pin C0..5), merupakan saluran masukan/keluaran dua arah dan juga mempunyai fungsi khusus. Fungsi khusus *Port C* adalah sebagai ADC (*input ADC channel 0..5*), selain itu pada beberapa *Port* memiliki fungsi lain antara

- lain Port C4 (SDA (*Two-Wire serial bus data input/output line*)), dan Port C5 (SCL (*Two-Wire serial bus clock line*)).
- Pin C6 memiliki fungsi *RESET*, merupakan saluran dua masukan untuk mereset mikrokontroler dengan cara memberi masukan logika rendah.
 - Port D (Pin D0..7), merupakan saluran masukan/keluaran dua arah dan juga mempunyai fungsi khusus. Fungsi khusus dari Port D diantaranya adalah : Port D0 {RXD (*USART input pin*)}, Port D1 {TXD (*USART output pin*)}, Port D2 {INT0 (*Eksternal interrupt 0 input*)}, Port D3 {INT1 (*Eksternal interrupt 1 input*)}, Port D4 (T0 (*timer/counter0 eksternal counter input*) & XCK (*USART eksternal clock input/output*)), Port D5 (T1 (*timer/counter eksternal counter input*)), Port D6 (AIN0 (*Analog comparator positive input*)) dan Port D7 (AIN1 (*Analog comparator negative input*)).
 - Pin 7 VCC, merupakan masukan untuk catu daya positif DC sebesar 5 volt.
 - Pin 8 GND, merupakan *ground* dari seluruh rangkaian.
 - Pin B6 dan Pin B7 (XTAL2 dan XTAL1), merupakan saluran untuk mengatur pewaktuan sistem. Untuk pewaktuan dapat menggunakan pewaktuan internal maupun eksternal.
 - Pin 21 AREF, merupakan pin referensi analog untuk masukan ADC.
 - Pin 22 GND, merupakan *ground* dari ADC.
 - Pin 20 AVCC, merupakan catu untuk perangkat ADC

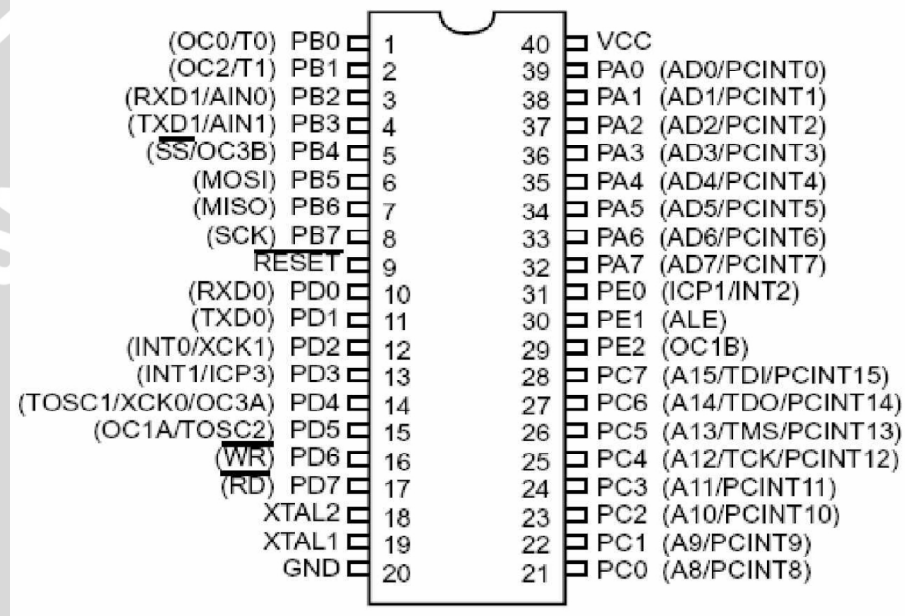
2.6.2 ATMEGA162

ATMega162 merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Hampir semua instruksi dieksekusi dalam satu siklus clock. ATMega162 mempunyai 32 register *general-purpose*, *timer/counter* fleksibel dengan mode *compare*, *interrupt* internal dan eksternal, *serial UART*, *programmable Watchdog Timer*, dan *mode power saving*. ATMega162 mempunyai ADC dan PWM internal. ATMega162 juga mempunyai *In-System Programmable Flash on-chip* yang mengijinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI. Atmega162 mempunyai *throughput* mendekati 1 MIPS per MHz membuat disainer sistem dapat mengoptimasi konsumsi daya versus kecepatan proses. Beberapa keistimewaan dari AVR ATMega162 antara lain:

- 1) *Advanced RISC Architecture*
 - *Most Single-clock Cycle Execution - 131 Powerful Instructions*
 - *General Purpose Working Registers (32 x 8)*
 - *Up to 16 MIPS Throughput at 16 MHz*
 - *On-chip 2-cycle Multiplier*
- 2) *Non-volatile Program and Data Memories*
 - *In-System Self-programmable Flash Endurance (16K Bytes): 1,000 Write/Erase Cycles*
 - *Optional Boot Code Section with Independent Lock Bits In-System Programming by On-chip Boot Program True Read-While-Write Operation*
 - *EEPROM Endurance (512 Bytes): 100,000 Write/Erase Cycles*
 - *Internal SRAM : 1K Bytes*
 - *Up to 64K Bytes Optional External Memory Space*
 - *Programming Lock for Software Security*
- 3) *Peripheral Features*
 - *Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes*
 - *Two 16-bit Timer/Counters with Separate Prescalers, Compare Modes, and Capture Modes*
 - *Real Time Counter with Separate Oscillator*
 - *Six PWM Channels*
 - *Dual Programmable Serial USARTs*
 - *Master/Slave SPI Serial Interface*
 - *Programmable Watchdog Timer with Separate On-chip Oscillator*
 - *On-chip Analog Comparator*
- 4) *Special Microcontroller Features*
 - *Power-on Reset and Programmable Brown-out Detection*
 - *Internal Calibrated RC Oscillator*
 - *External and Internal Interrupt Sources*
 - *Five Sleep Modes: Idle, Power-save, Power-down, Standby, and Extended Standby*
- 5) *I/O and Packages*
 - *Programmable I/O Lines (35)*
 - *PDIP (40-pin), 44-lead TQFP, and 44-pad MLF*
- 6) *Operating Voltages*

- ATmega162V : 1.8 - 3.6V
- ATmega162L : 2.7 - 5.5V
- ATmega162 : 4.5 - 5.5V

Untuk memaksimalkan performa dan paralelisme, ATmega162 menggunakan arsitektur Harvard (memori dan bus terpisah untuk program dan data). Gambar 2.19 menunjukkan pin-pin pada ATmega162 dengan kemasan 40-pin DIP (*Dual In-Line Package*).



Gambar 2.19. Pin-Pin pada ATmega162 dengan Kemasan 40-Pin DIP (*Dual In-Line Package*)
 Sumber: Atmel, 2009: 2

Adapun fungsi dari pin-pin yang terdapat pada Mikrokontroler ATmega162 dijelaskan sebagai berikut :

- a) VCC, suplai tegangan digital
- b) GND, pin *ground*
- c) PORT A (PA0 – PA7). PORT A merupakan port I/O 8-bit *bidirectional* yang dilengkapi dengan resistor *pull-up* internal (dapat dipilih untuk tiap *bit*). Selain sebagai port I/O, PORT A juga mempunyai fungsi lain seperti antarmuka memori eksternal dan pin *change interrupt*.
- d) PORT B (PB0 – PB7). PORT B merupakan port I/O 8-bit *bidirectional* dengan resistor *pull-up* internal (dapat dipilih untuk tiap *bit*). Selain itu, PORT B juga

- mempunyai fungsi lain yaitu *Serial Peripheral Interface (SPI)*, *Analog Comparator*, *input/output Timer/Counter*.
- e) PORT C (PC0 – PC7). PORT C merupakan port I/O 8-bit *bidirectional* dengan resistor *pull-up* internal (dapat dipilih untuk tiap *bit*). Selain itu PORT C juga mempunyai fungsi lain yaitu JTAG, antarmuka memori eksternal, dan pin *change interrupt*.
 - f) PORT D (PD0 – PD7). PORT D merupakan port I/O 8-bit *bidirectional* dengan resistor *pull-up* internal (dapat dipilih untuk tiap *bit*). Selain itu PORT D juga mempunyai fungsi lain yaitu USART, *interupsi eksternal*, *strobe memori eksternal*, *timer/counter*.
 - g) PORT E (PE0 – PE2) PORT E merupakan port I/O 3-bit *bidirectional* dengan resistor *pull-up* internal (dapat dipilih untuk tiap *bit*). Selain itu PORT E juga mempunyai fungsi lain yaitu *timer/counter*, *latch enable memori eksternal*, *interupsi eksternal*.
 - h) RESET, berfungsi untuk mereset mikrokontroler jika diberikan sinyal *active low* dalam selang waktu tertentu.
 - i) XTAL1, input ke inverting *oscillator* amplifier dan ke rangkaian detak internal.
 - j) XTAL2, output dari inverting *oscillator amplifier*.

Universal synchronous dan *asynchronous* pemancar dan penerima *serial* adalah suatu alat komunikasi serial sangat fleksibel. Mikrokontroler ATmega162 memiliki dua buah *port* USART untuk komunikasi *serial*, yaitu USART0 dan USART1. Fasilitas komunikasi serial USART mikrokontroler ini memiliki fitur sebagai berikut:

- 1) *Operasi full duplex* (register penerima dan pengirim *serial* dapat berdiri sendiri)
- 2) *Operasi Asynchronous* atau *synchronous*
- 3) *Master* atau *slave* mendapat *clock* dengan operasi *synchronous*
- 4) Pembangkit *baud rate* dengan resolusi tinggi
- 5) Dukung *frames serial* dengan 5, 6, 7, 8 atau 9 *data bit* dan 1 atau 2 *stop bit*
- 6) Tahap *odd* atau *even parity* dan *parity check* didukung oleh *hardware*
- 7) Pendeteksian data *overrun*
- 8) Pendeteksi *framing error*

- 9) Pemfilteran gangguan (*noise*) meliputi pendeteksian *bit false start* dan pendeteksian *low pass filter* digital
- 10) Tiga *interrupt* terdiri atas *TX complete*, *TX data register empty*, dan *RX complete*
- 11) Mode komunikasi *multi-processor*
- 12) Mode komunikasi *double speed asynchronous*

2.7 Program CodeVision AVR

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

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

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

IDE mempunyai fasilitas internal berupa *software AVR Chip In-System Programmer* yang memungkinkan pengguna untuk melakukan transfer program kedalam chip mikrokontroler setelah sukses melakukan kompilasi/assembly secara otomatis. *Software In-System Programmer* didesain untuk bekerja dengan Atmel STK500/AVRISP/AVRProg, Kanda Systems STK200+/300, Dontronics DT006, Vogel Elektronik VTEC-ISP, Futurlec JRAVR dan MicroTronics ATCPU/Mega2000 *programmers/development boards*.

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

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

- a) Modul LCD alphanumeric
- b) Bus I2C dari Philips
- c) Sensor Suhu LM75 dari National Semiconductor

- repository.ub.ac
- d) *Real-Time Clock*: PCF8563, PCF8583 dari Philips, DS1302 dan DS1307 dari Maxim/Dallas Semiconductor
 - e) Protokol 1-Wire dari Maxim/Dallas Semiconductor
 - f) Sensor Suhu DS1820, DS18S20, dan DS18B20 dari Maxim/Dallas Semiconductor
 - g) Termometer/Termostat DS1621 dari Maxim/Dallas Semiconductor
 - h) EEPROM DS2430 dan DS2433 dari Maxim/Dallas Semiconductor
 - i) SPI
 - j) *Power Management*
 - k) Delay
 - l) Konversi ke Kode Gray

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

- a) *Set-up* akses memori eksternal
- b) Identifikasi sumber *reset* untuk *chip*
- c) Inisialisasi port *input/output*
- d) Inisialisasi interupsi eksternal
- e) Inisialisasi *Timer/Counter*
- f) Inisialisasi *Watchdog-Timer*
- g) Inisialisasi UART (USART) dan komunikasi *serial* berbasis *buffer* yang digerakkan oleh interupsi
- h) Inisialisasi Pembanding Analog
- i) Inisialisasi ADC
- j) Inisialisasi Antarmuka SPI
- k) Inisialisasi Antarmuka *Two-Wire*
- l) Inisialisasi Antarmuka CAN
- m) Inisialisasi Bus I2C, Sensor Suhu LM75, *Thermometer/Thermostat* DS1621 dan *Real-Time Clock* PCF8563, PCF8583, DS1302, dan DS1307
- n) Inisialisasi *Bus 1-Wire* dan Sensor Suhu DS1820, DS18S20
- o) Inisialisasi modul LCD

2.8 Rotary Encoder

Sensor *Rotary encoder* yang digunakan pada skripsi ini merupakan produksi dari Autonics dengan tipe E40H8-500P/R-6-L-5 seperti dalam Gambar 2.16. Sensor ini memiliki lubang dengan diameter 80mm sebagai tempat untuk mengkopel antara sensor dengan roda. Keluaran dari sensor ini berupa pulsa dengan tegangan $5V \pm 5\%$ saat pulsa *high* dan $0V$ saat pulsa *low*. Sensor ini dapat menyediakan 500 pulsa/ rotasi. Pulsa keluaran dari sensor ini dapat diartikan menjadi gerakan, posisi, dan arah sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder* untuk diteruskan oleh rangkaian kendali.



Gambar 2.20 Rotary Encoder
Sumber :autonics, 2013

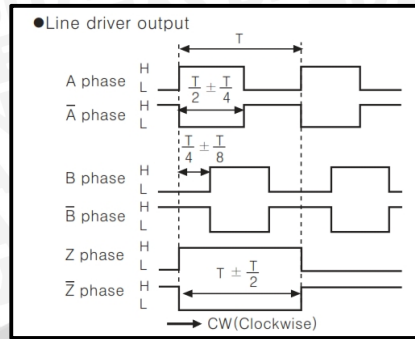
Sensor ini mempunyai 9 kaki atau pin. Konfigurasi dari masing- masing pin ditunjukkan dalam Tabel 2.3.

Tabel 2.3 Konfigurasi Pin *Rotary Encoder*

No Pin	Function	Warna Kabel
1	OUT A	Hitam
2	OUT A'	Merah
3	+V	Coklat
4	GND	Biru
5	OUT B	Putih
6	OUT B'	Abu – abu
7	OUT Z	Jingga
8	OUT Z'	Kuning
9	F.G	Pelindung kebel

Sumber : Datasheet Autonics Rotary Encoder

Perbedaan antara masing-masing keluaran dari sensor *rotary encoder* ini dapat kita lihat dalam Gambar 2.21.



Gambar 2.21 Line Driver Output Sensor Rotary Encoder
 Sumber : Dataheet autonics rotary encoder

Dalam Gambar 2.17 diketahui bahwa terdapat pergeseran fasa antara OUT A dan OUT B dengan periode pulsa yang sama, sedangkan OUT Z memiliki periode 2 kali lebih lama. Fasa yang berkebalikan juga disediakan oleh rotary ini pada OUT A', B' dan Z'.



BAB III

METODOLOGI

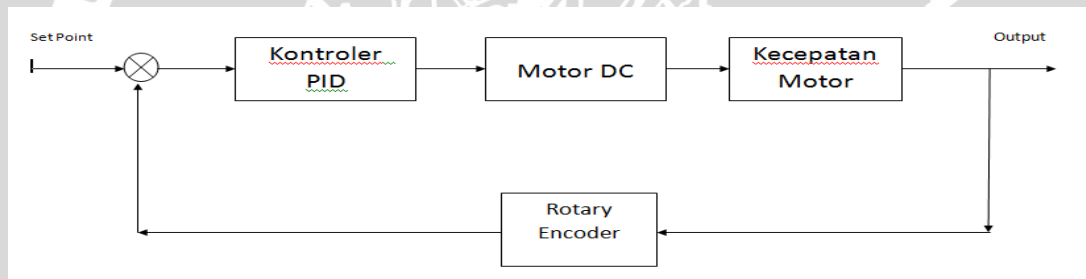
Untuk merealisasikan alat yang telah dirancang, langkah-langkah yang dilakukan adalah sebagai berikut:

3.1 Studi literatur

Studi literatur dilakukan agar dapat memahami komponen penyusun sistem. Studi literatur yang dilakukan tentang *rotary encoder*, kontroler PID, motor DC, *atmega8*, *atmega16* dan *software cvavr*.

3.2 Perancangan Blok Diagram Sistem

Pada perancangan alat diperlukan perancangan blok diagram sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan rencana.



Gambar 3.1 Blok Diagram Sistem (Sumber:Perancangan)

3.3 Prinsip Kerja

Saat robot diaktifkan bersamaan dengan tombol ON, sistem dari alat ini bekerja sesuai dengan sensor cahaya yang mendeteksi garis hitam dan putih, pada kondisi ini robot masih dalam keadaan persiapan (*prepare*). Saat tombol start diaktifkan, baru robot aktif secara keseluruhan, robot digerakkan oleh dua motor DC magnet permanen. Sensor *photodiode* membaca garis hitam dan mengirimkan sinyal input ke- mikrokontroler sensor ATMEGA8 untuk selanjutnya diteruskan ke ATMEGA16. Mikrokontroler ATMEGA16 sebagai MK utama berfungsi mengolah data dari MK sensor (MK1), MK motor kiri (MK2) serta MK motor kanan (MK3) menggunakan *kontroler PID*. Kemudian MK utama mengirimkan perintah pengaturan kecepatan motor ke mikrokontroler ATMEGA8 untuk motor kiri dan mikrokontroler ATMEGA8 untuk motor kanan. Pengaturan kecepatan motor

dilakukan oleh MK2 dan MK3 berdasar perintah dari MK utama. Saat tombol start aktif tersebut, pengaturan kecepatan motor dilakukan oleh mikrokontroller yang memperoleh data dari *rotary encoder* sebagai sensor kecepatan, untuk mempertahankan kecepatan motor pada nilai yang diinginkan walaupun dengan beban yang berbeda-beda.

3.4 Perancangan Perangkat Keras

1. Membuat chassis robot dari bahan aluminium
2. Membuat body robot
3. Memasang motor penggerak ke *chassis* robot
4. Membuat rangkaian elektrik robot dan memasangkannya ke *body* robot
5. Membuat rangkaian mekanik tempat sampah
6. Membuat lintasan robot

3.5 Perancangan Perangkat Lunak

Untuk perancangan perangkat lunak hal pertama yang dilakukan adalah mengetahui karakteristik motor DC, setelah itu dilakukan perhitungan untuk menentukan algoritma kontroler dengan Kontroler PID. Setelah didapatkan algoritma, kemudian dibuat program untuk mikrokontroler dengan *software CodeVisionAVR C Compiler*.

3.6 Pengujian dan Analisis Sistem

Untuk memastikan sistem ini berjalan dengan baik maka perlu dilakukan pengujian sistem, meliputi pengujian perangkat keras (*hardware*) maupun perangkat lunak (*software*) yang dilakukan baik secara blok rangkaian maupun keseluruhan sistem.

3.7 Pengambilan Kesimpulan dan Saran

Kesimpulan diambil berdasarkan data yang diperoleh dari pengujian sistem secara keseluruhan. Sedangkan saran diberikan untuk pengembangan skripsi ini.

BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

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

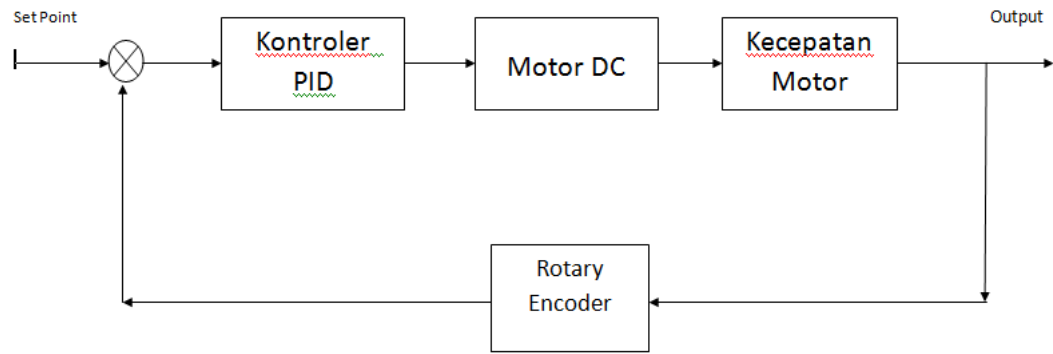
4.1 Spesifikasi Alat

Spesifikasi alat yang dirancang adalah sebagai berikut:

1. Robot *Line Follower* menggunakan rangka dengan spesifikasi sebagai berikut:
 - Dimensi robot secara keseluruhan yaitu panjang 100 cm, lebar 40 cm dan tinggi 45 cm.
 - Bak robot terbuat dari acrylic dengan panjang 70 cm, lebar 40 cm dan tinggi 40 cm
2. Pergerakan Robot *Line Follower Berbeban* menggunakan dua buah motor DC.
3. Dua buah *Rotary Encoder* sebagai sensor kecepatan yang dipasang pada masing-masing motor DC.
4. Menggunakan 8 (delapan) buah sensor *photo reflective*, enam berada di bagian depan dan satu di pojok belakang samping kanan dan kiri.
5. Mikrokontroler yang digunakan adalah satu buah ATMEGA 16 dan tiga buah ATMEGA 8
6. Catu tegangan untuk mikrokontroler menggunakan baterai *Li-po* 11.1 V 2200mA.
7. Catu tegangan untuk motor DC menggunakan *accu* 12 V.

4.2 Diagram Blok Sistem

Diagram blok sistem ditunjukkan dalam Gambar 4.1



Gambar 4.1 Blok Diagram Sistem (Sumber: Perancang)

Keterangan diagram blok dalam Gambar 4.1 adalah:

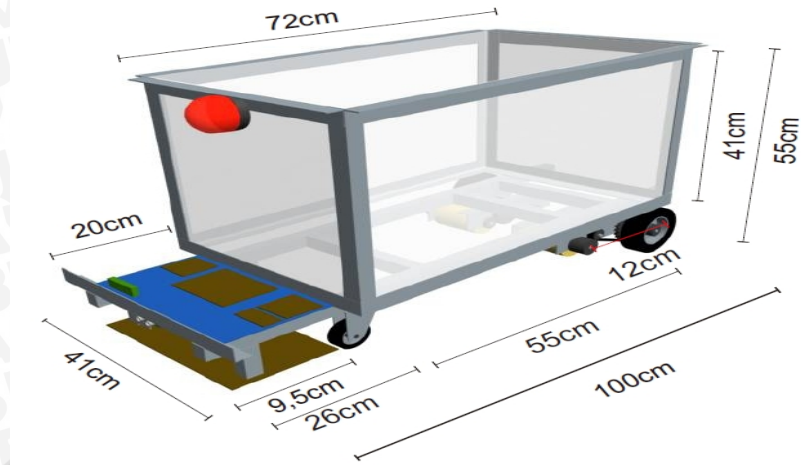
- *Input* berupa nilai kecepatan motor DC .
- Sinyal *feedback* dari sensor rotary encoder berupa pembacaan putaran motor DC.
- *Error* berupa selisih antara nilai setpoint dengan nilai pembacaan sensor.
- Kontroler yang digunakan adalah Proporsional, Integral, dan Diferensial dengan menggunakan mikrokontroler untuk mengolah nilai PID. Nilai PID digunakan sebagai nilai referensi pemanggilan nilai kontrol kecepatan motor DC.
- Aktuator berupa 2 (dua) motor DC.
- *Plan* berupa kecepatan motor pada robot *line follower* berbeban.
- *Output* berupa nilai kecepatan motor yang sesuai dengan nilai *setpoint*.

4.3 Perancangan Perangkat Keras

Perangkat keras terdiri dari rangka *robot line follower* berbeban, board mikrokontroler, *rotary encoder*, dan motor DC.

4.3.1 Perancangan Rangka Robot *Line Follower* Berbeban

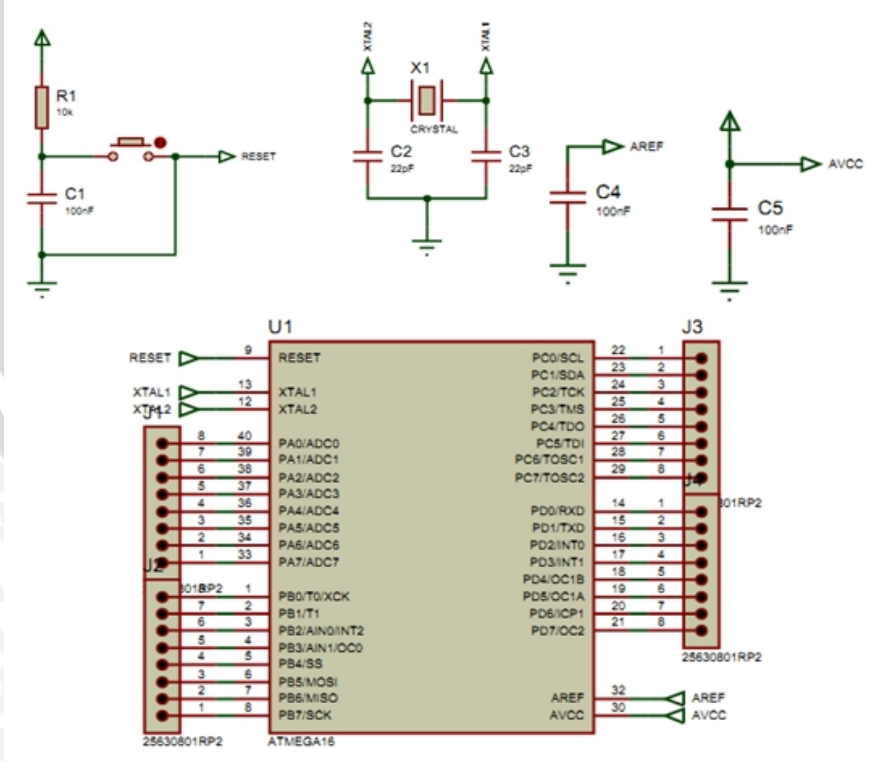
Perancangan rangka dilakukan sebagai dasar dari komponen-komponen yang akan diletakkan. Secara umum rangka robot *line follower* berbeban ditunjukkan dalam Gambar 4.2.



Gambar 4.2 Rangka Robot Line Follower Berbeban (Sumber: Perancangan.)

4.3.2 Rangkaian Board Mikrokontroler

Pada robot ini digunakan mikrokontroler ATmega16 sebagai penerima data pembacaan sensor garis dari Mikrokontroler Slave Sensor dan mengirimkan data ke Mikrokontroler Slave Motor Kanan dan Mikrokontroler Slave Motor Kiri untuk melakukan perintah terhadap aktuator (motor DC). Konfigurasi kaki I/O dalam mikrokontroler ATmega16 (mikrokontroler *master*) ditunjukkan dalam Gambar 4.3

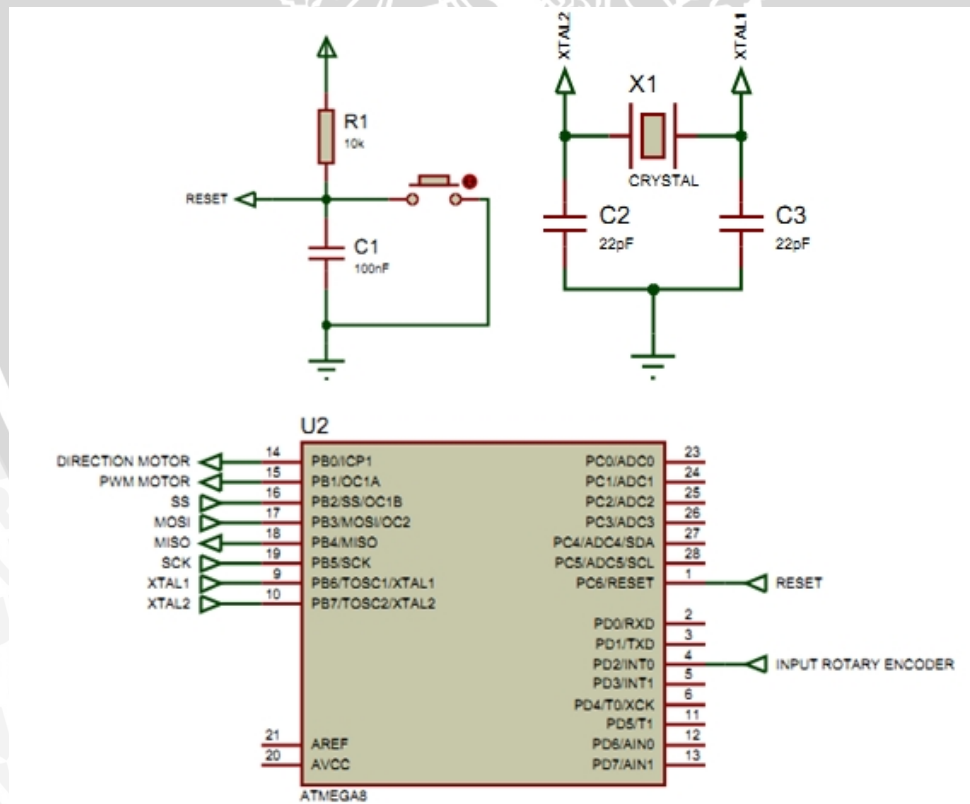


Gambar 4.3 Rangkaian Minimum Sistem Mikrokontroler Master (Sumber: Perancangan)

Rangkaian minimum sistem mikrokontoller master yang digunakan mempunyai 4 port, dengan 32 jalur I/O yang dapat di program menjadimasukanataukeluaran. Pin dan kaki IC yang digunakan antara lain:

- Pin B5(MOSI) : Sebagai pin keluaran dataSPI.
- Pin B6(MISO) : Sebagai pin masukan data SPI.
- Pin B7(SCK) : Sebagai pin keluaran clock SPI.
- Pin Reset : Dihubungkan dengan rangkaian switch Reset.
- Pin XTAL1 : Dihubungkandengan kaki osilator 11059200 Hz.
- Pin XTAL2 : Dihubungkandengan kaki osilator 11059200 Hz.
- Pin AREF : Dihubungkan dengan rangkaian AREF.
- Pin AVCC : Dihubungkan dengan rangkaian AVCC.

Pada robot ini juga digunakan mikrokontroler ATMega8 sebagai pengolah data rotary encoder. Konfigurasi kaki I/O dari mikrokontroler ATMega8 (mikrokontroler *slave*) ditunjukkan dalam Gambar 4.4



Gambar 4.4 Rangkaian Minimum Sistem Mirokontroler *Slave* Motor
(Sumber: Perancangan)

Mikrokontroler ATmega8 mempunyai 3 port, dengan 21 jalur I/O yang dapat di program menjadi masukan atau keluaran. Pin yang digunakan antara lain:

- Pin B0 : Sebagai pin keluaran Direction Motor.
- Pin B1(OC1A) : Sebagai pin keluaran PWM Motor.
- Pin B2(MOSI) : Sebagai pin masukan Clock SPI.
- Pin B3(MOSI) : Sebagai pin masukan data SPI.
- Pin B4(MISO) : Sebagai pin keluaran data SPI.
- Pin B5(SCK) : Sebagai pin masukan clock SPI.
- Pin D2(INT0) : Sebagai pin masukan Rotary Encoder.
- Pin Reset : Dihubungkan dengan rangkaian switch Reset.
- Pin XTAL1 : Dihubungkan dengan kaki osilator 11059200 Hz.
- Pin XTAL2 : Dihubungkan dengan kaki osilator 11059200 Hz

4.3.3 Rotary Encoder

Dibutuhkan dua Rotary Encoder yang dicouple langsung dengan masing-masing motor DC. *Rotary Encoder* berfungsi sebagai sensor kecepatan. *Rotary Encoder* yang digunakan ditunjukkan dalam Gambar 4.5.



Gambar 4.5 Rotary Encoder
(Datashett Rotary Encoder Autonics)

4.3.4 Pemilihan Motor DC

Motor DC yang digunakan adalah motor DC Nisca 5475 dengan rpm 1930 pada tegangan 12 V. Motor DC yang digunakan ditunjukkan pada gambar 4.5.



Gambar 4.5 Motor DC dan Rangkaian Gear box (Sumber: Perancangan)

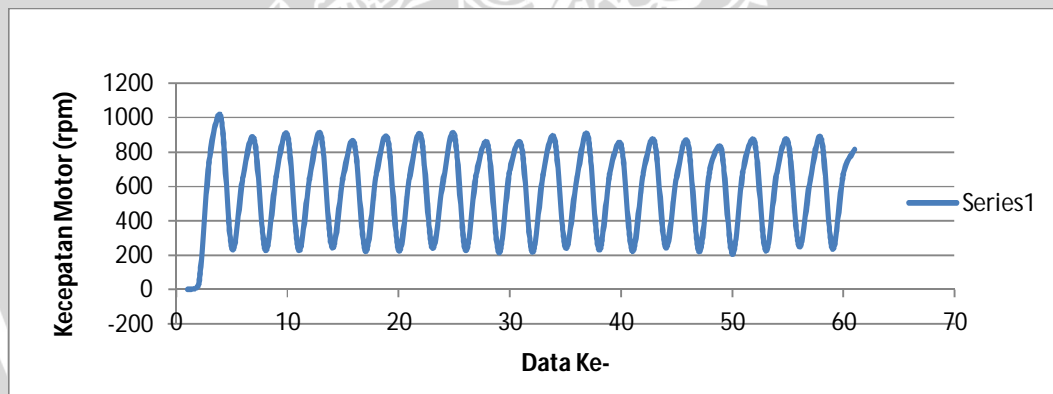
4.4 Perancangan Sistem Tuning Kontroler PID Menggunakan Metode Kedua Ziegler-Nichols

Proses Tuning PID pada perancangan ini menggunakan metode kedua Ziegler Nichols sehingga harus dicari terlebih dahulu nilai K_p saat terjadi osilasi berkesinambungan pada masing – masing motor dengan nilai K_i dan K_d adalah 0. Dari nilai K_p saat terjadi osilasi berkesinambungan tersebut dapat dicari K_{cr} dan P_{cr} . Dengan nilai K_{cr} dan P_{cr} , dapat dihitung nilai K_i dan K_d .

Nilai *set point* (kecepatan putaran roda yang diinginkan) yang akan dipakai harus ditentukan terlebih dahulu Pada perancangan ini ditetapkan set point yang digunakan sebesar 600 rpm untuk memperoleh parameter PID.

4.4.1 Perancangan *Tuning* Parameter PID Pada Motor 1 (Kanan)

Proses *Tuning* diawali dengan merubah nilai K_p dari 1-10. Grafik respon kecepatan motor yang mempunyai osilasi berkesinambungan didapat saat nilai kontroler proporsional 5 ($K_{cr}=5$). Hasil pengujian kecepatan motor dengan menggunakan kontroler proporsional dengan nilai 5 ($K_{cr}=5$) dapat dilihat dalam Gambar 4.6



Gambar 4.6 Grafik respon kecepatan roda 1 dengan $K_p=5$, $K_i=0$, $K_d=0$ ($K_{cr} = 5$)

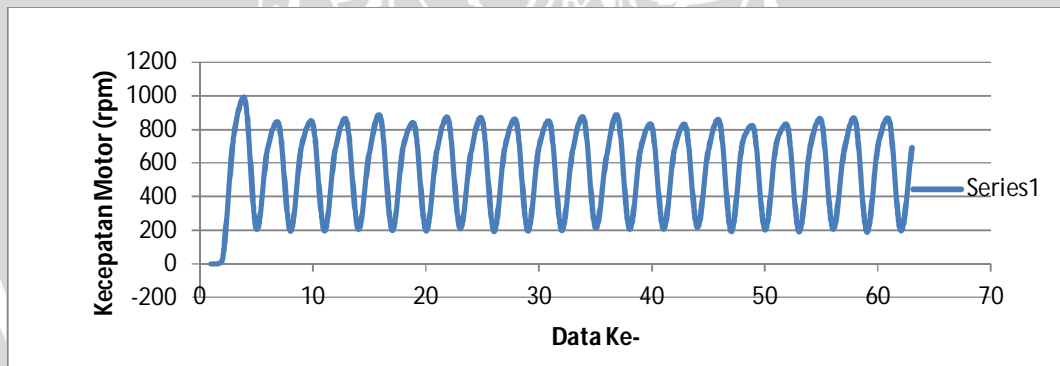
Gambar 4.6 menunjukkan bahwa pada saat kontroler proporsional bernilai 5 dapat membentuk osilasi berkesinambungan. Respon sistem menampilkan data setiap 0,6 s sehingga nilai K_{cr} dan P_{cr} dapat dihitung. Nilai K_i dan K_d dapat diperoleh dengan perhitungan sebagai berikut:

- $K_{cr} = 5$
- $P_{cr} = 0,6$ sekon
- $K_p = 0,6 \times K_{cr} = 0,6 \times 5 = 3$
- $T_i = 0,5 \times P_{cr} = 0,5 \times 0,6 = 0,3$
- $T_d = 0,125 \times P_{cr} = 0,125 \times 0,6 = 0,075$
- $K_i = \frac{K_p}{T_i} = \frac{3}{0,3} = 10$
- $K_d = K_p \times T_d = 3 \times 0,075 = 0,225$

Hasil *tuning* parameter PID dengan menggunakan metode kedua Ziegler-Nichols pada motor 1 diperoleh nilai $K_p = 3$, $K_i = 10$, dan $K_d = 0,225$

4.4.2 Perancangan *Tuning* Parameter PID Pada Motor 2 (Kiri)

Proses *Tuning* diawali dengan merubah nilai K_p dari 1-10. Grafik respon kecepatan motor yang mempunyai osilasi berkesinambungan didapat saat kontroler proporsional bernilai 4,9 ($K_{cr}=4,9$). Hasil pengujian respon kecepatan motor dengan menggunakan kontroler proporsional dengan nilai 4,9 ($K_{cr}=4,9$) dapat dilihat dalam Gambar 4.7.



Gambar 4.7 Grafik respon kecepatan motor 2 dengan $K_p=4,9$, $K_i=0$, $K_d=0$ ($K_{cr} = 4,9$)

Gambar 4.7 menunjukkan bahwa pada saat kontroler proporsional bernilai 4,0 kecepatan motor dapat membentuk osilasi berkesinambungan. Respon sistem menampilkan data setiap 0,6 s sehingga nilai k_{cr} dan P_{cr} dapat dihitung. Nilai K_i dan K_d dapat diperoleh dengan perhitungan sebagai berikut:

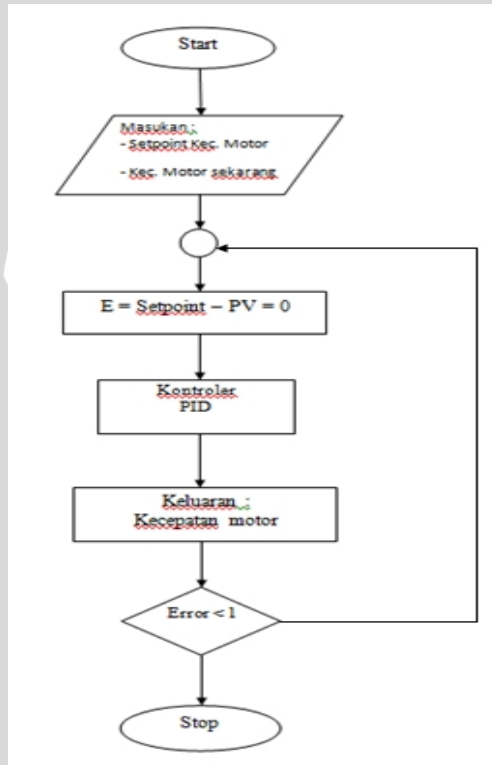
- $K_{cr} = 4,9$
- $P_{cr} = 0,6$ sekon
- $K_p = 0,6 \times K_{cr} = 0,6 \times 4,9 = 2,94$
- $T_i = 0,5 \times P_{cr} = 0,5 \times 0,6 = 0,3$

- $Td = 0,125 \times Pcr = 0,125 \times 0,6 = 0,075$
- $Ki = \frac{Kp}{Ti} = \frac{2,94}{0,3} = 9,8$
- $Kd = Kp \times Td = 2,94 \times 0,075 = 0,2205$

Hasil *tuning* parameter PID dengan menggunakan metode kedua Ziegler-Nichols pada motor 2 diperoleh nilai $Kp = 2,94$, $Ki = 9,8$, dan $Kd = 0,2205$

4.4 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada skripsi ini menggunakan bahasa pemrograman C++ dengan menggunakan *software* CodeVision AVR. *Tuning* kontroler PID adalah dengan menggunakan metode Ziegler-Nichols. *Flowchart* perancangan perangkat lunak ditunjukkan dalam Gambar 4.8.



Gambar 4.8 Flowchart Perangkat Lunak (Perancangan, 2013)

BAB V

PENGUJIAN DAN ANALISIS SISTEM

Pengujian dan analisis dilakukan untuk mengetahui apakah sistem dapat bekerja sesuai perancangan yang telah dilakukan. Pengujian dilakukan per blok sistem kemudian secara keseluruhan. Adapun pengujian yang dilakukan adalah sebagai berikut:

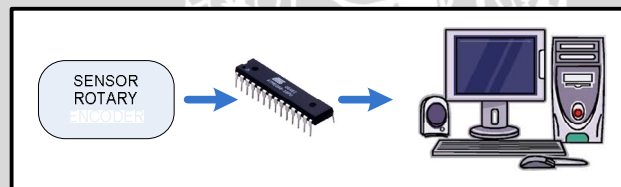
- 1) Pengujian sensor *Rotary Encoder*
- 2) Pengujian respon kontroler PID
- 3) Pengujian keseluruhan system

5.1 Pengujian Sensor Rotary Encoder

Pengujian ini bertujuan untuk melihat apakah sensor rotary encoder yang dipakai dapat menghasilkan pulsa sesuai dengan datasheet dari rotary encoder dan untuk melihat apakah rotary encoder dapat berfungsi sebagai sensor kecepatan roda.

5.1.1 Pengujian Keluaran Sensor Rotary Encoder

Pada datasheet, rotary encoder dapat menghasilkan 500 pulsa dalam satu putaran. Prosedur pengujian dilakukan dengan menghubungkan antara rotary encoder, mikrokontroler ATmega8 dan komputer sesuai dengan Gambar 5.1



Gambar 5.1 Diagram Blok Pengujian Keluaran Sensor Rotary Encoder (Sumber: Perancangan)

Pada pengujian ini, sensor rotary encoder diputar sebesar $\frac{1}{4}$, $\frac{1}{2}$, $\frac{3}{4}$, dan 1 putaran. Pada masing – masing putaran dilihat berapa pulsa yang terbaca oleh mikrokontroler. Jumlah pulsa yang terbaca ditampilkan pada komputer. Hasil pengujian yang diperoleh dari beberapa pengambilan data ditunjukkan pada Tabel 5.1

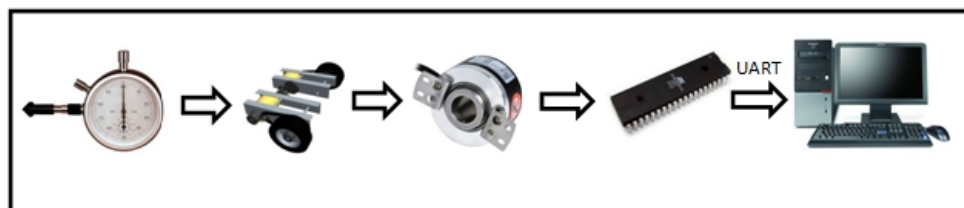
Tabel 5.1 Hasil Pengujian data pulsa rotary encoder

Pengujian ke-	Besar putaran	Jumlah pulsa rotary 1	Jumlah pulsa rotary 2
1	0,25	125 125 126	125 125 125
2	0,5	247 250 248	250 250 251
3	0,75	375 375 375	373 374 375
4	1,00	500 501 502	500 501 498

Berdasarkan Tabel 5.1 dapat diperoleh hasil bahwa kesalahan rata – rata yang terjadi saat pembacaan jumlah pulsa rotary encoder sebesar 0.75 pulsa atau dibulatkan menjadi 1 pulsa. Kesalahan pembacaan terbesar yaitu 3 pulsa. Pada pengujian, kesalahan pembacaan berupa jumlah pulsa yang kurang atau lebih besar dari nilai seharusnya. Kesalahan ini dapat terjadi karena kesalahan dalam memutar rotary encoder. Dari Pengujian ini terlihat bahwa keluaran dari *rotary encoder* dapat dibaca oleh mikrokontroler dan pulsa keluaran dari rotary encoder sesuai dengan *datasheet*

5.1.2 Pengujian Rotary Encoder Sebagai Sensor Kecepatan

Pengujian dilakukan dengan menghubungkan mikrokontroler *slaved* dengan rotary encoder dan komputer melalui UART. Pengukuran kecepatan motor yang sebenarnya menggunakan tachometer analog yang dikopel langsung dengan motor. Diagram blok pengujian ditunjukkan dalam Gambar 5.2



Gambar 5.2 Diagram Blok Pengujian Rotary Encoder Sebagai Sensor Kecepatan (Sumber: Perancangan)

Pengujian dilakukan dengan memberikan nilai PWM tertentu. Kecepatan motor diukur dengan menggunakan rotary encoder dan tachometer. Hasil pengukuran kecepatan motor dengan rotary encoder ditampilkan pada komputer. Hasil pengujian ditunjukkan dalam Tabel 5.2

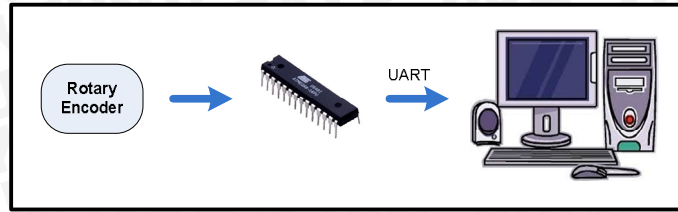
Tabel 5.2 Hasil Pengujian Kecepatan motor dengan Rotary Encoder Dan Tachometer

Rotary	PWM (rpm)	Rotary (rpm)	Tacho (rpm)	Kesalahan (%)	Kesalahan rata-rata	Kec. roda (rpm)
motor 1(kiri)	25%	714	708	0,85	1,23	122
	50%	1305	1310	0,38		215
	75%	1450	1400	3,57		240
	100%	1558	1560	0,13		293
motor 2(kanan)	25%	825	810	1,85	1,17	136
	50%	1280	1270	0,79		215
	75%	1422	1420	0,14		236
	100%	1513	1485	1,89		250

Dari Tabel 5.2 diperoleh kesalahan rata – rata pada motor 1 sebesar 1,23% dan motor 2 sebesar 1,17%. Rotary encoder dapat berfungsi sebagai sensor kecepatan motor dengan kesalahan rata-rata sebesar 1,2%.

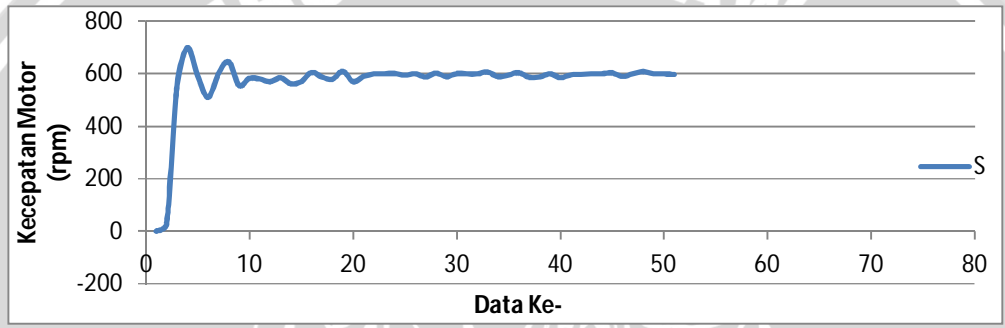
5.2 Pengujian Hasil *Tuning* Parameter PID

Tujuan dari pengujian ini adalah untuk melihat respon roda dari nilai parameter kontrol PID (Kp, Ki dan Kd) dengan menggunakan metode Osilasi Ziegler-Nichols yang telah didapat sebelumnya. Prosedur pengujian dilakukan dengan memasukkan nilai parameter yang telah didapat ke persamaan PID pada masing – masing *slave*. Hubungkan mikrokontroler slave ke komputer untuk melihat data kecepatan putaran motor dari sensor rotary encoder yang dikopel langsung dengan roda. Diagram blok pengujian hasil tuning parameter kontroler PID menggunakan metode osilasi Ziegler-Nichols dalam satu sistem mikrokontroler *slave* ditunjukkan dalam gambar 5.3.



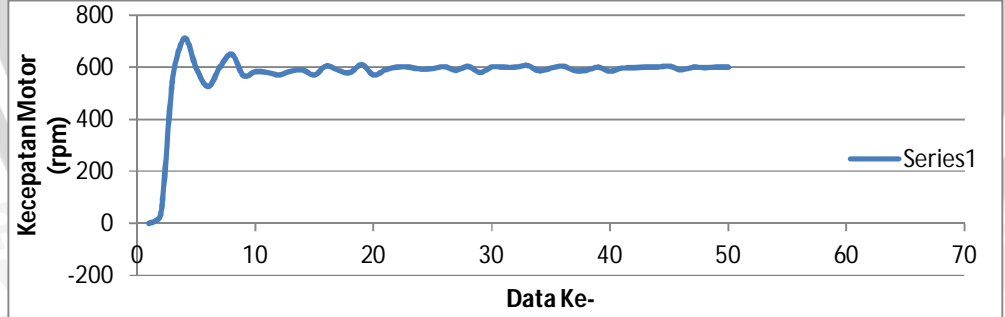
Gambar 5.3 Diagram Blok Pengujian Hasil *Tuning* Parameter Kontroler PID.
(Sumber: Perancangan)

Dari data yang diterima komputer dapat dilihat grafik respon kecepatan putaran roda. Hasil *tuning* parameter kontroler PID menggunakan metode kedua Ziegler – Nichols pada motor 1 didapat nilai $K_p = 3$, $K_i = 10$, dan $K_d = 0,225$. Grafik respon kecepatan pada motor 1 ditunjukkan dalam Gambar 5.4.



Gambar 5.4 Grafik Respon Kecepatan Motor 1

Hasil *tuning* parameter kontroler PID pada roda 2 menggunakan metode kedua Ziegler – Nichols didapat nilai $K_p = 2,94$, $K_i = 9,8$ dan $K_d = 0,2205$. Grafik respon kecepatan motor 2 ditunjukkan dalam Gambar 5.5.



Gambar 5.5 Grafik Respon Kecepatan Motor 2

5.3 Pengujian Keseluruhan Sistem

Pengujian ini bertujuan untuk mengetahui keberhasilan dari keseluruhan sistem yang dirancang.pada pengujian ini semua sistem digabungkan menjadi satu

kesatuan dan menggunakan kontrol PID sebagai sistem kontrolnya. Pengujian dilakukan untuk mengetahui kontrol sistem terhadap beban yang berubah-ubah.

5.3.1 Peralatan Pengujian

Peralatan yang dibutuhkan antara lain:

1. Robot *Line Follower* Pembawa Beban
2. Sensor *rotary encoder*
3. Mikrokontroler Atmega 8
4. Komputer
5. Beban dan timbangan

5.3.2 Prosedur Pengujian

Prosedur pengujiannya adalah:

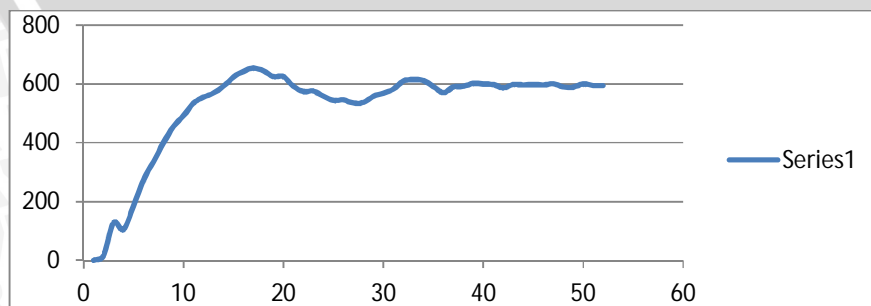
1. Menyiapkan dan menimbang beban yang akan digunakan
2. Meletakkan beban pada box robot
3. Menjalankan system
4. Melihat respon sistem yang dikirim ke *computer* melalui *software* Proteus

5.3.3 Hasil Pengujian

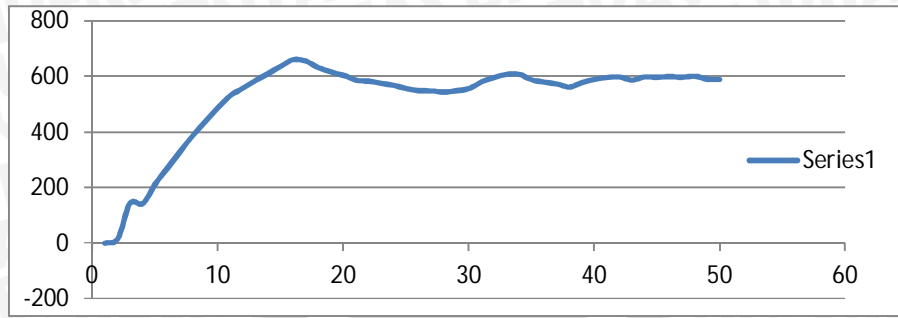
Setelah sistem dijalankan dengan dibebani beban yang berbeda-beda. Maka kita dapat mengetahui respon yang dihasilkan oleh system. Variasi beban yang digunakan adalah sebagai berikut :

1. Beban box robot
2. Beban box robot dengan tambahan beban 2,5 kg
3. Beban box robot dengan tambahan beban 5 kg
4. Beban box robot dengan tambahan beban 7,5 kg

Respon sistem dengan beban box robot (tanpa beban tambahan) ditunjukkan dalam Gambar 5.6 untuk motor 1 dan Gambar 5.7 untuk motor 2.

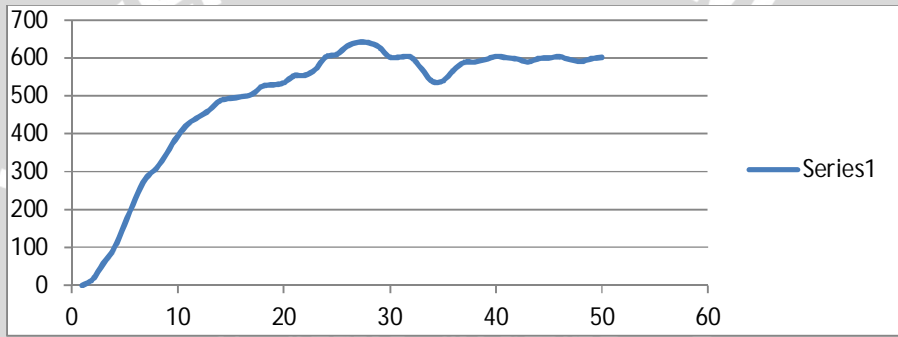


Gambar 5.6 Respon Motor 1 dengan beban box menggunakan kontrol PID

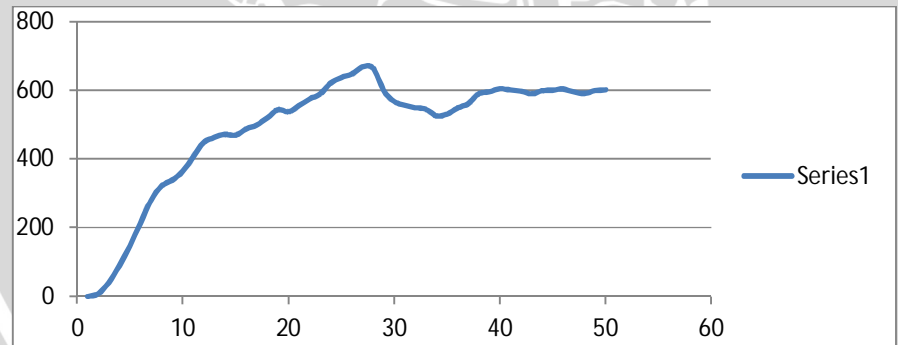


Gambar 5.7 Respon Motor 2 dengan beban box menggunakan kontrol PID

Respon system dengan box dan beban 2,5 kg ditunjukkan dalam Gambar 5.8 untuk motor 1 dan Gambar 5.9 untuk motor 2.

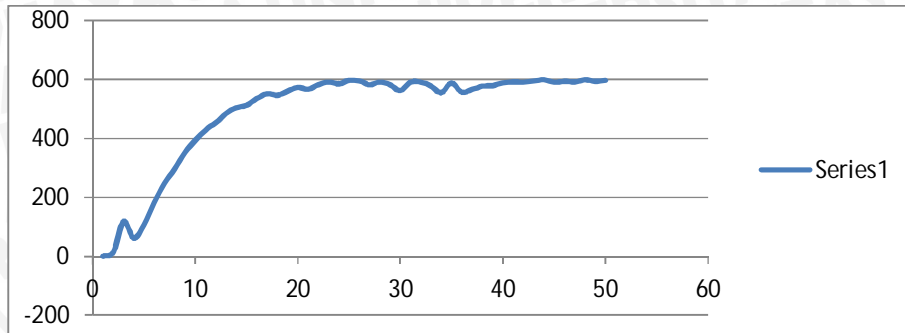


Gambar 5.8 Respon Motor 1 dengan box dan beban 2,5 kg menggunakan kontrol PID

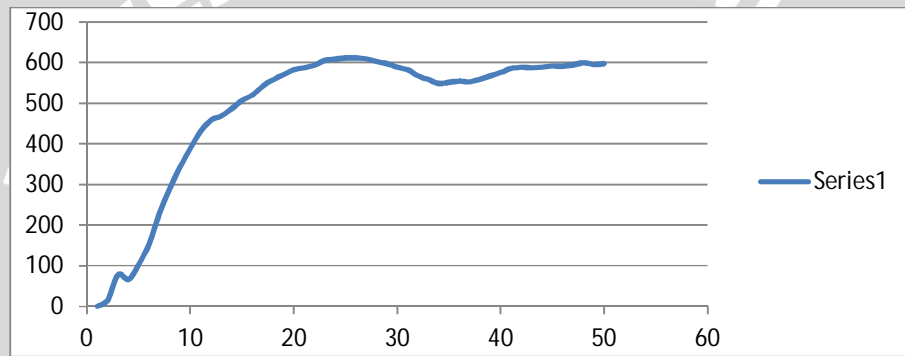


Gambar 5.9 Respon Motor 2 dengan box dan beban 2,5 kg menggunakan kontrol PID

Respon system dengan box dan beban 5 kg ditunjukkan dalam Gambar 5.10 untuk motor 1 dan Gambar 5.11 untuk motor 2.

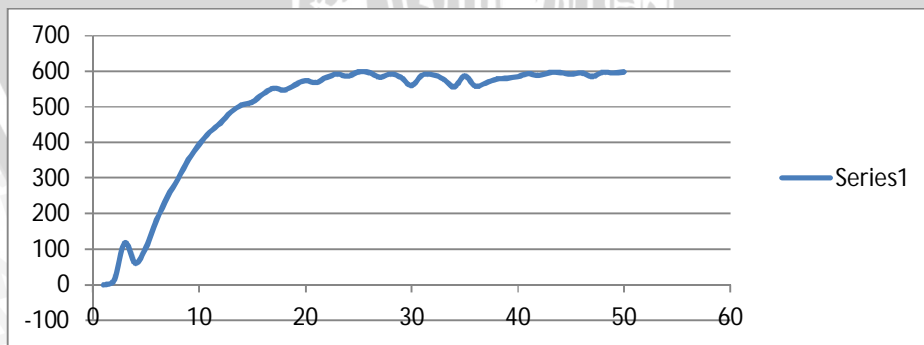


Gambar 5.10 Respon Motor 1 dengan box dan beban 5 kg menggunakan kontrol PID

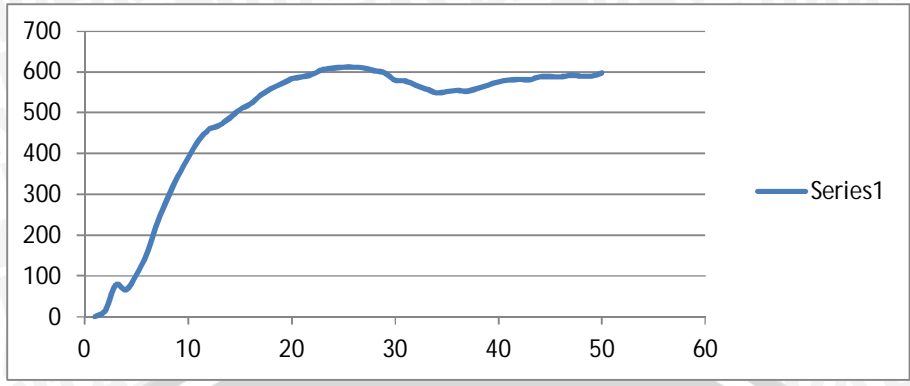


Gambar 5.11 Respon Motor 2 dengan box dan beban 5 kg menggunakan kontrol PID

Respon system dengan box dan beban 7,5 kg ditunjukkan dalam Gambar 5.12 untuk motor 1 dan Gambar 5.13 untuk motor 2.



Gambar 5.12 Respon Motor 1 dengan box dan beban 7,5 kg menggunakan kontrol PID



Gambar 5.13 Respon Motor 2 dengan box dan beban 7,5 kg menggunakan kontrol PID

Pengujian ini ditentukan memiliki presentase kesalahan yang diperbolehkan adalah 5% berdasarkan sistem yang dinyatakan..

Tabel 5.3 Tabel Data Performansi Sistem

<u>Beban</u>	Motor	<u>Settling time (ts)</u>	<u>Maximum Overshoot (%)</u>	<u>Ess (%)</u>
Box (5kg)	Motor 1	6,2 detik	9	0,88
	Motor 2	6,4detik	10	1,375
Box+ 2,5 kg	Motor 1	7,4 detik	6,8	0,67
	Motor 2	7,4detik	11	1
Box+ 5 kg	Motor 1	7,6 detik	0	1,3
	Motor 2	8 detik	0,3	1,5
Box + 7,5 kg	Motor 1	8,2 detik	0	1,16
	Motor 2	8,8 detik	0,16	1,66

Dari Tabel 5.3 dapat disimpulkan bahwa system bekerja dengan baik dengan rata-rata error steady state sebesar 1,19%.

5.4 Pengujian Beban Yang Tidak Seimbang Antara Motor Kanan dan Motor Kiri

Pengujian ini bertujuan untuk mengetahui keberhasilan dari keseluruhan sistem yang dirancang dengan keadaan beban yang tidak seimbang antara motor kanan dan motor kiri. Pada pengujian ini semua sistem digabungkan menjadi satu kesatuan dan menggunakan kontrol PID sebagai sistem kontrolnya. Pengujian dilakukan untuk mengetahui perbedaan respon motor dan performa robot secara keseluruhan ketika dikenakan beban yang tidak seimbang antara motor kanan dan motor kiri.

5.4.1 Peralatan Pengujian

Peralatan yang dibutuhkan antara lain:

1. Robot *Line Follower* Pembawa Beban
2. Sensor *rotary encoder*
3. Mikrokontroler Atmega 8
4. Komputer
5. Beban dan timbangan

5.4.2 Prosedur Pengujian

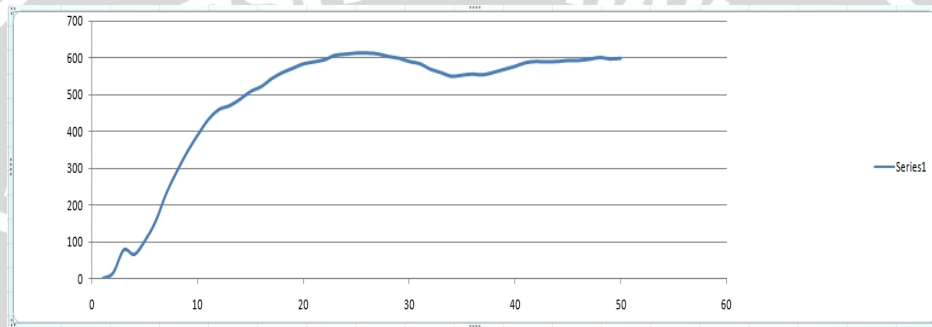
Prosedur pengujiannya adalah:

1. Menyiapkan dan menimbang beban yang akan digunakan yaitu seberat 7,5 Kg
2. Meletakkan beban pada box robot secara tidak seimbang antara sisi kanan dan sisi kiri
3. Menjalankan system
4. Melihat respon sistem yang dikirim ke *computer* melalui *software* Proteus

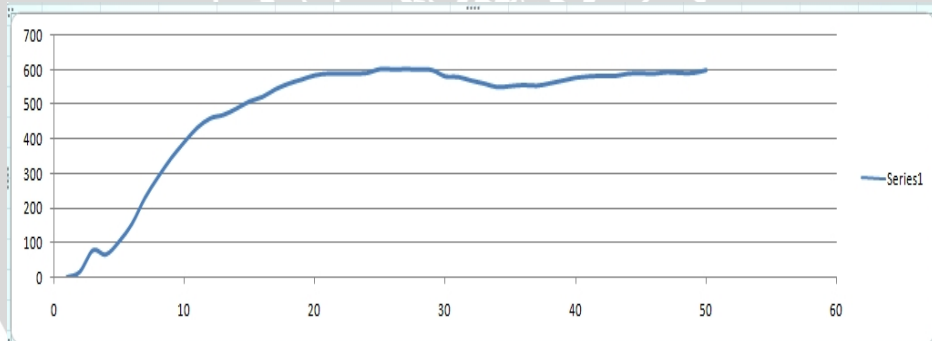
5.4.3 Hasil Pengujian

Setelah dilakukan pengujian dengan keadaan beban *unbalance* dapat diamati bahwa performa robot tidak mengalami masalah. Sistem masih dapat mengontrol kecepatan motor pada set poin yang telah ditentukan dan Robot tetap bisa mengikuti garis dengan baik. Dapat disimpulkan bahwa sensor *rotary encoder* dan sensor cahaya bekerja dengan baik.

Perbedaan respon system ketika sisi kanan robot lebih berat (*unbalance*) dapat dilihat pada grafik respon di bawah.



Gambar 5.14 Respon motor kanan ketika sisi kanan robot lebih berat dari pada sisi kiri robot



Gambar 5.15 Respon motor kiri ketika sisi kanan robot lebih berat dari pada sisi kiri robot

BAB VI PENUTUP

6.1 Kesimpulan

Dari hasil perancangan dan pengujian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

1. Dengan menggunakan kontroler PID yang telah dirancanag menggunakan metode Ziegler Nichols, robot *line follower* berbeban ini telah mampu mempertahankan kecepatan motor dengan beban yang berbeda-beda sesuai dengan set point yang sudah ditetapkan yaitu sebesar 600 rpm. Rata-rata error *steady state* dari pengujian yang telah dilakukan adalah sebesar 1,19%. Parameter PID yang diperoleh adalah $K_p=3$, $K_i=10$, $K_d=0,225$ untuk motor 1 dan $K_p=4,9$, $K_i=9,8$, $K_d=0,2205$ untuk motor 2.
2. Dalam penelitian ini respon yang dihasilkan kontroler masih bisa dikatakan lambat, yaitu rata-rata diperlukan waktu 6 second untuk mencapai *steady sate*.
3. Sensor *rotary encoder* berfungsi dengan baik sebagai sensor kecepatan dengan *error* kesalahan sebesar 1,2%.
4. Pemilihan motor DC Nisca 5475 dapat bekerja dengan baik pada tegangan 12 V dengan toleransi beban yang telah ditentukan.

6.2 Saran

Beberapa hal yang direkomendasikan untuk pengembangan lebih lanjut adalah:

1. Disarankan untuk penelitian selanjutnya agar mencoba menggunakan metode kontroler yang lain untuk mencapai respon yang lebih cepat dari sebelumnya.
2. Disarankan untuk menngunakan motor dengan torsi yang lebih besar agar mampu membawa beban yang lebih berat.
3. Penambahan load sensor untuk mengetahui berat beban yang dibawa oleh robot.

DAFTAR PUSTAKA

Amstrom, K. J., & Hagglund, Tore. 1995 *PID Controllers: Theory, Design and Tuning*. Instrument Society of America: Research Triangle Park.

Atmel Corporation. 2006. *ATMEGA 5835 Series*.

Autonics. 2013. Rotary Encoder Datasheet. Yangan: Autonics Corporation

Dwi Septian Suyadhi, Taufiq. 2010. *Buku Pintar Robotika Bagaimana Merancang & Membuat Robot Sendiri*. Yogyakarta: ANDI.

Fahmizal. 2010. *Robot Line Follower dengan Kendali PID*.
<http://fahmizaleeits.wordpress.com/2010/05/08/robot-line-follower-dengan-kendali-pid/> diakses tanggal 1 Desember 2013

Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik (Sistem Pengaturan)*. Jakarta: Erlangga.

