

**SISTEM PENGENDALI KECEPATAN MOTOR DC PADA *TOWER CRANE* MENGGUNAKAN KONTROLER PID BERBASIS MIKROKONTROLER**

**SKRIPSI**

**TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL**

Diajukan untuk Memenuhi Persyaratan  
Memperoleh Gelar Sarjana Teknik



**Disusun oleh:**

**RAINIER LESTIANTO**

**NIM. 105060302111002**

**UNIVERSITAS BRAWIJAYA**

**FAKULTAS TEKNIK**

**MALANG**

**2016**

**LEMBAR PENGESAHAN**  
**SISTEM PENGENDALI KECEPATAN MOTOR DC PADA TOWER**  
**CRANE DENGAN MENGGUNAKAN KONTROLER PID**  
**BERBASIS MIKROKONTROLER**

**SKRIPSI**

**TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL**

Ditujukan untuk memenuhi persyaratan  
 memperoleh gelar Sarjana Teknik



**RAINIER LESTIANTO**  
**NIM. 105060302111002**

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing  
 Pada tanggal 18 Mei 2016

Dosen Pembimbing I

Ir. Purwanto, MT.  
 NIP. 19540424 198601 1 001

Dosen Pembimbing II

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

Mengetahui

Ketua Jurusan Teknik Elektro

M. Aziz Muslim, S.T., M.T., Ph.D.  
 NIP. 19741203 200012 1 001



**JUDUL SKRIPSI:**

**SISTEM PENGENDALI KECEPATAN MOTOR DC PADA *TOWER CRANE* DENGAN MENGGUNAKAN KONTROLER PID BERBASIS MIKROKONTROLER**

**Nama Mahasiswa : RAINIER LESTIANTO**

**NIM : 105060302111002**

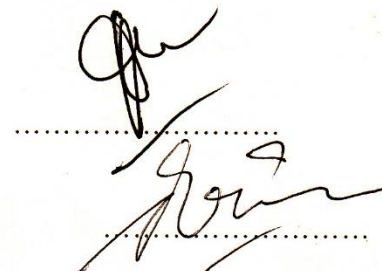
**Program Studi : Teknik Elektro**

**Konsentrasi : Teknik Kontrol**

**Komisi Pembimbing :**

**Ketua : Ir. Purwanto, MT**

**Anggota : Goegoes Dwi Nusantoro, ST.,MT.**



**TIM DOSEN PENGUJI :**

**Dosen Penguji 1 : Dr. Ir. Erni Yudaningtyas, MT**

**Dosen Penguji 2 : M. Aziz Muslim, ST., MT., Ph.D**

**Dosen Penguji 3 : Ir. Retnowati, M.T.**



**Tanggal Ujian : 13 Mei 2016**

**SK Penguji : No.591/UN10.6/SK/2016**

**PERNYATAAN ORISINALITAS SKRIPSI**

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 18 Mei 2016

Mahasiswa,

(Materai Rp 6.000,-)

(Tanda Tangan)

Rainier Lestianto

NIM. 105060302111002



## KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa sehingga saya dapat menyelesaikan penyusunan makalah ini dalam bentuk maupun isinya yang sangat sederhana. Semoga skripsi ini dapat dipergunakan sebagai salah satu acuan, petunjuk maupun pedoman bagi pembaca dalam pendidikan. Sungguh hanya melalui Pertolongan dan Perlindungan Tuhan Yang Maha Esa semata sehingga saya dapat menyelesaikan skripsi ini. Dengan seizin Tuhan Yang Maha Esa, di kesempatan yang baik ini saya ingin menghaturkan rasa terima kasih dan penghargaan yang sebesar besarnya atas bantuan sehingga terselesainya skripsi ini kepada:

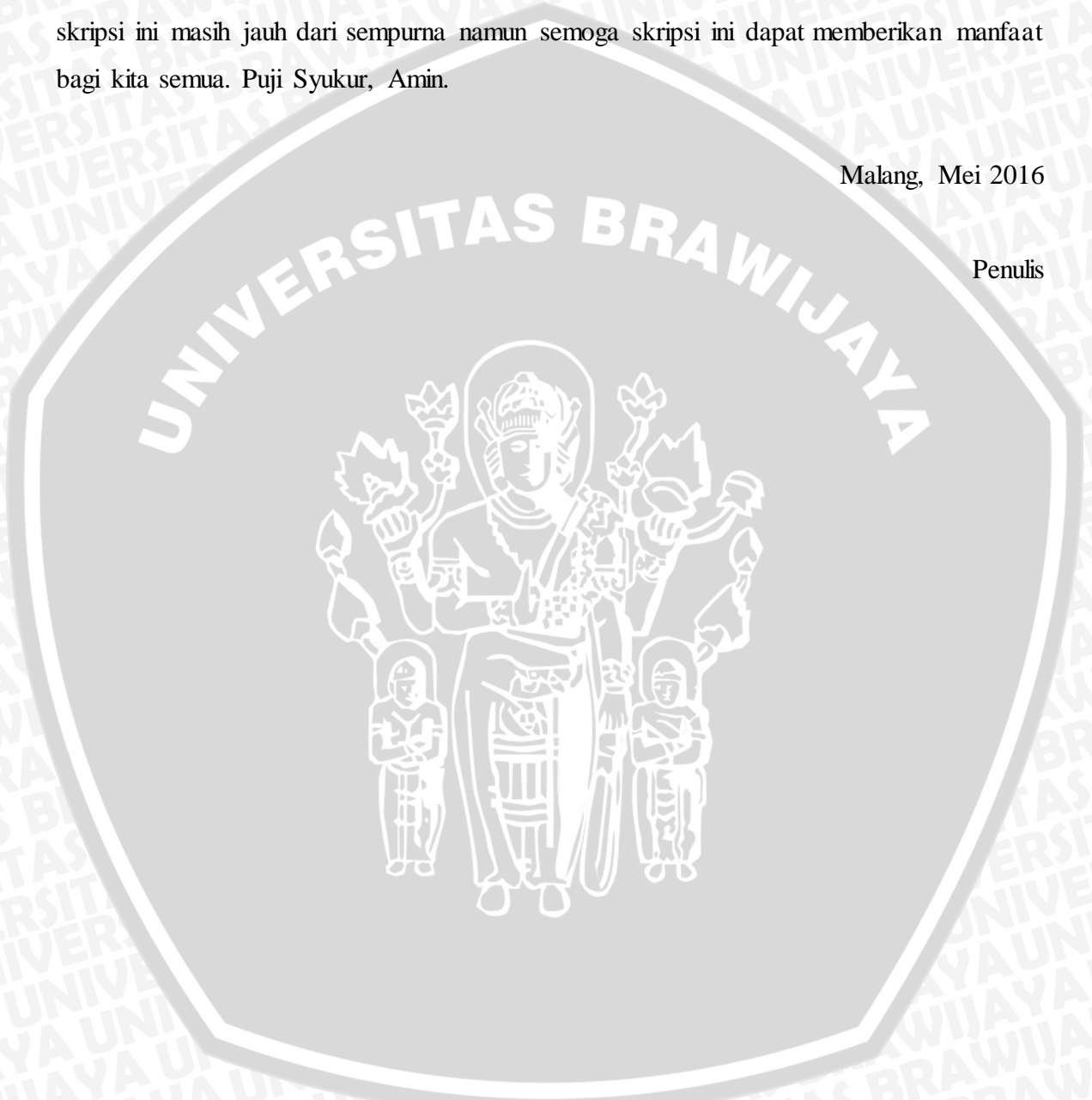
- Keluarga tercinta, kedua orang tua Eddy Lestianto dan Sandrana Kainde yang selalu memberikan kasih sayang dan doanya yang tiada akhir. Serta kakak Gerald Lestianto yang selalu memberikan dukungan.
- Bapak M. Aziz Muslim, ST., MT., Ph.D selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Hadi Suyono, ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Ir. Purwanto, MT. selaku KKDK Teknik Kontrol dan sebagai Dosen Pembimbing I atas segala bimbingan, pengarahan, gagasan, ide, saran, motivasi yang telah diberikan, serta waktu yang diluangkan untuk bimbingan.
- Bapak Goegoes Dwi Nusantoro, ST., MT. sebagai Dosen Pembimbing II atas segala bimbingan, pengarahan, gagasan, ide, saran, motivasi yang telah diberikan, serta waktu yang diluangkan untuk bimbingan.
- Bapak Ibu Dosen, karyawan, staf recording dan RBTE atas segala bantuan dan kemudahan.
- Teman - teman “MaGiC”, Luthfi, Gilang, Hanip, Rangga, Mukson, Randy, Dandy, Ulit, Dany, Dugal, Ernanda, Hanip, Rowman, Smith dan Kadek terima kasih telah berbagi kesenangan, pelajaran hidup, serta canda dan tawa.

- Keluarga besar angkatan 2010 MAGNET'10 atas do'a, semangat, serta dukungan yang diberikan pada penulis.
- Semua pihak yang telah memberikan bantuan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Sekiranya Tuhan Yang Maha Esa mencatat amalan ikhlas kami dan semua pihak yang turut membantu sehingga skripsi ini terselesaikan. Akhirnya, kami menyadari bahwa skripsi ini masih jauh dari sempurna namun semoga skripsi ini dapat memberikan manfaat bagi kita semua. Puji Syukur, Amin.

Malang, Mei 2016

Penulis



## ABSTRAK

**Rainier Lestianto**, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Mei 2016, Sistem Pengendali Kecepatan Motor DC Pada *Tower Crane* Menggunakan Kontroler PID Berbasis Mikrokontroler, Dosen Pembimbing: Ir. Purwanto, MT., Goegoes Dwi Nusantoro, ST.,MT.

*Tower Crane* berfungsi sebagai alat angkat bahan - bahan dan material seperti mesin-mesin konstruksi, beton, besi, bekisting dan lain sebagainya. Cara kerja *Tower Crane* adalah dengan memindahkan barang bangunan secara vertikal dan horizontal. *Tower Crane* banyak digunakan untuk proyek-proyek pembangunan gedung-gedung bertingkat dan juga jembatan. Pengendalian pada *Tower Crane* umumnya dikerjakan oleh tenaga kerja manusia. Tenaga kerja manusia yang digunakan juga harus memiliki kemampuan khusus dan terlatih agar dapat mengendalikan *Tower Crane* secara aman. Hal ini sangatlah tidak efisien karena berat barang yang akan dipindahkan sangatlah beragam. Pada penelitian ini telah dirancang sistem pengaturan kecepatan motor DC pada *Tower Crane* menggunakan kontroler PID dengan konstruksi sistem yang sederhana. Digunakan Kontroler PID untuk mengurangi kesalahan, sehingga putaran motor dapat sesuai dengan kecepatan yang diinginkan. Dengan bantuan kontroler PID maka *Tower Crane* mampu bergerak dengan aman dan halus. Pada skripsi ini digunakan metode pertama *Ziegler-Nichols*. Dalam pembuatannya digunakan Arduino Mega 2560, rotary encoder Autonic E40H8 500-6-L-5, optocoupler, dan motor DC. Berdasarkan data respon sistem yang diperoleh dari pengujian dengan menggunakan metode pertama *Ziegler-Nichols*, maka parameter kontroler PID dapat ditentukan dengan gain  $K_p = 0,091$ ,  $K_i = 0,003834$  dan  $K_d = 0,02172$ .

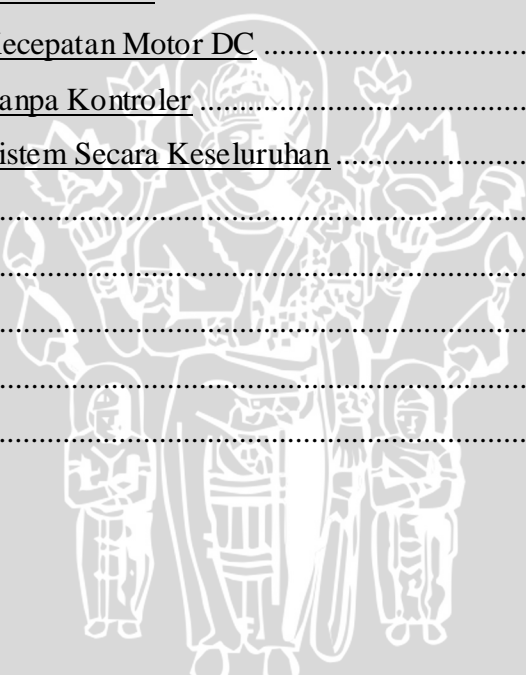
**Kata kunci:** *Tower Crane, Pengendalian Kecepatan, PID, Arduino Mega 2560.*

## DAFTAR ISI

<b><u>KATA PENGANTAR</u></b> .....	i
<b><u>ABSTRAK</u></b> .....	iii
<b><u>DAFTAR ISI</u></b> .....	iv
<b><u>DAFTAR GAMBAR</u></b> .....	vi
<b><u>DAFTAR TABEL</u></b> .....	viii
<b><u>BAB I PENDAHULUAN</u></b> .....	1
1.1 <u>Latar Belakang</u> .....	1
1.2 <u>Rumusan Masalah</u> .....	2
1.3 <u>Batasan Masalah</u> .....	2
1.4 <u>Tujuan</u> .....	3
1.5 <u>Sistematika Penulisan</u> .....	3
<b><u>BAB II TINJAUAN PUSTAKA</u></b> .....	5
2.1 <u>Pengertian Tower Crane</u> .....	5
2.2 <u>Arduino Mega 2560</u> .....	7
2.3 <u>Motor DC</u> .....	8
2.4 <u>Pulse Width Modulation (PWM)</u> .....	9
2.5 <u>Sensor Rotary Encoder</u> .....	10
2.6 <u>Optocoupler</u> .....	11
2.7 <u>Kontroler</u> .....	12
2.8 <u>Metode Perancangan Kontroler Proporsional Integral Diferensial (PID)</u> .....	16
2.9 <u>Driver motor DC L298n Dual H-Bridge</u> .....	19
<b><u>BAB III METODOLOGI PENELITIAN</u></b> .....	21
3.1 <u>Studi Literatur</u> .....	21
3.2 <u>Penentuan Spesifikasi Alat</u> .....	21
3.3 <u>Perancangan Sistem</u> .....	22
3.4 <u>Realisasi Pembuatan Sistem</u> .....	22
3.4.1 <u>Perangkat Keras</u> .....	22
3.4.2 <u>Perangkat Lunak</u> .....	22
3.5 <u>Pengujian Alat</u> .....	22
3.6 <u>Pengambilan Kesimpulan dan Saran</u> .....	23
<b><u>BAB IV PERANCANGAN DAN PEMBUATAN ALAT</u></b> .....	25
4.1 <u>Diagram Blok Sistem</u> .....	25



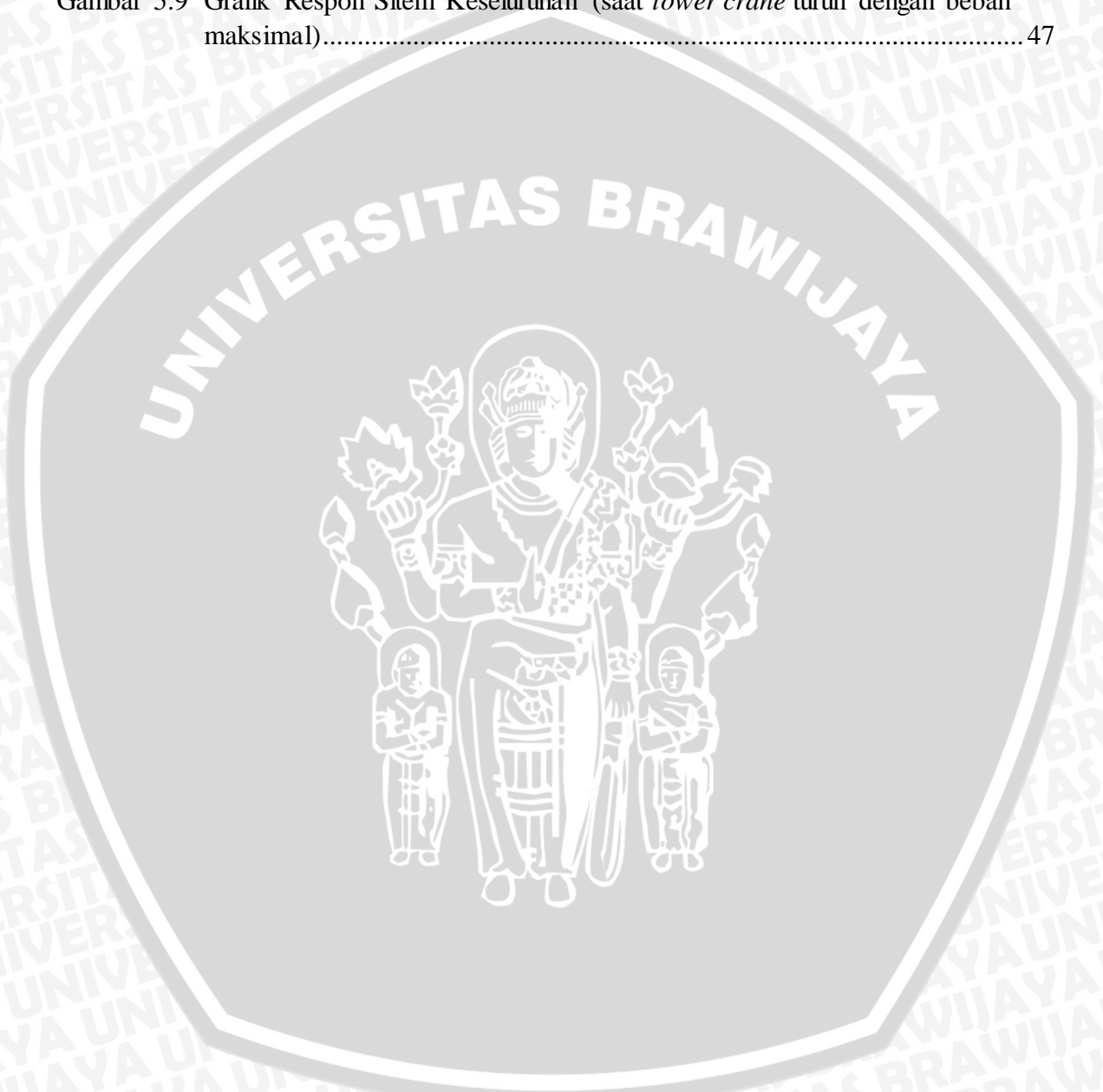
4.2	<u>Prinsip Kerja Alat</u> .....	26
4.3	<u>Perancangan Perangkat Keras</u> .....	26
4.3.1	<u>Spesifikasi Alat</u> .....	27
4.3.2	<u>Perancangan Driver Motor DC</u> .....	29
4.3.3	<u>Motor DC</u> .....	30
4.3.4	<u>Modul Arduino Mega 2560</u> .....	31
4.3.5	<u>Perancangan Kontroler</u> .....	32
4.3.6	<u>Perancangan Kontroler PID metode pertama Ziegler-Nichols</u> .....	33
4.4	<u>Perancangan Perangkat Lunak</u> .....	34
4.4.1	<u>Flowchart Sistem Keseluruhan</u> .....	34
<b><u>BAB V PENGUJIAN DAN ANALISIS SISTEM</u></b> .....		37
5.1	<u>Pengujian Sensor Rotary Encoder</u> .....	37
5.2	<u>Pengujian Driver Motor</u> .....	39
5.3	<u>Pengujian Kecepatan Motor DC</u> .....	41
5.4	<u>Pengujian Tanpa Kontroler</u> .....	42
5.5	<u>Pengujian Sistem Secara Keseluruhan</u> .....	43
<b><u>BAB VI PENUTUP</u></b> .....		49
6.1	<u>Kesimpulan</u> .....	49
6.2	<u>Saran</u> .....	49
<b><u>DAFTAR PUSTAKA</u></b> .....		51
<b><u>LAMPIRAN</u></b> .....		53



## DAFTAR GAMBAR

Gambar 2.1	<i>Tower Crane</i> .....	5
Gambar 2.2	Arduino Mega 2560 Tampak Atas.....	7
Gambar 2.3	Cara Kerja Motor DC.....	9
Gambar 2.4	Sinyal PWM Secara Umum .....	10
Gambar 2.5	Diagram <i>Rotary Encoder</i> .....	10
Gambar 2.6	<i>Optocoupler</i> .....	11
Gambar 2.7	Diagram Blok Kontroler Proporsional.....	13
Gambar 2.8	Diagram Blok Kontroler Integral.....	14
Gambar 2.9	Diagram Blok Kontroler Diferensial .....	14
Gambar 2.10	Diagram Blok Kontroler PID .....	16
Gambar 2.11	Kurva Respon <i>Unit Step</i> yang Menunjukkan 25% <i>Maximum Overshoot</i> ...	16
Gambar 2.12	Respon <i>Plant</i> Terhadap Masukan Berupa Unit Step .....	17
Gambar 2.13	Kurva Respon Yang Berbentuk S .....	17
Gambar 2.14	Sistem Loop Tertutup Dengan Kontroler <i>Proporsional</i> .....	18
Gambar 2.15	Osilasi Berkesinambungan Dengan Periode Pcr.....	18
Gambar 2.16	Module driver L298N Dual H-Bridge .....	19
Gambar 4.1	Blok Diagram Sistem.....	25
Gambar 4.2	Rangka Model Miniatur.....	27
Gambar 4.3	<i>Rotary Encoder</i> Sejajar Dengan Motor DC .....	28
Gambar 4.4	Model Miniatur <i>Tower Crane</i> .....	28
Gambar 4.5	Penempatan Sensor <i>Optocoupler</i> .....	29
Gambar 4.6	Koneksi Modul L298N <i>Dual H-Bridge</i> .....	29
Gambar 4.7	Modul Rangkaian <i>Driver L298N Dual H-Bridge</i> .....	30
Gambar 4.8	Grafik Karakteristik Motor DC.....	30
Gambar 4.9	Desain Sistem Arduino Mega 2560 .....	31
Gambar 4.10	Grafik Respon Sistem Tanpa Kontroler.....	32
Gambar 4.11	Metode 1 Ziegler-Nichols (Perancangan).....	33
Gambar 4.12	<i>Flowchart</i> Keseluruhan Sistem.....	35
Gambar 5.1	Grafik Perbandingan Nilai Pembacaan Sensor dengan <i>Tachometer</i> .....	39
Gambar 5.2	Grafik Hubungan PWM Dengan Tegangan Keluaran <i>Driver</i> .....	40
Gambar 5.3	Grafik Hubungan PWM Dengan Kecepatan Motor .....	42

Gambar 5.4	Pengujian Sistem tanpa kontroler dengan Masukan <i>Unit Step</i> .....	43
Gambar 5.5	Model Miniatur <i>Tower Crane</i> Secara Keseluruhan .....	44
Gambar 5.6	Grafik Respon Sitem Keseluruhan (saat <i>tower crane</i> naik tanpa beban) ....	45
Gambar 5.7	Grafik Respon Sitem Keseluruhan (saat <i>tower crane</i> turun tanpa beban) ...	46
Gambar 5.8	Grafik Respon Sitem Keseluruhan (saat <i>tower crane</i> naik dengan beban maksimal).....	46
Gambar 5.9	Grafik Respon Sitem Keseluruhan (saat <i>tower crane</i> turun dengan beban maksimal).....	47



DAFTAR TABEL

Tabel 2.1 Kecepatan Dan Kapasitas Berat *Tower Crane* ..... 6

Tabel 2.2 Spesifikasi *Tower Crane* Tipe JL4210A..... 6

Tabel 2.3 Aturan Penalaan *Ziegler-Nichols* Berdasarkan Respon Unit Step Dari Plan 17

Tabel 2.4 Aturan Dasar *Ziegler-Nichols* Berdasarkan *Critical Gain Kcr* Dan *Critical Period Pcr* ..... 18

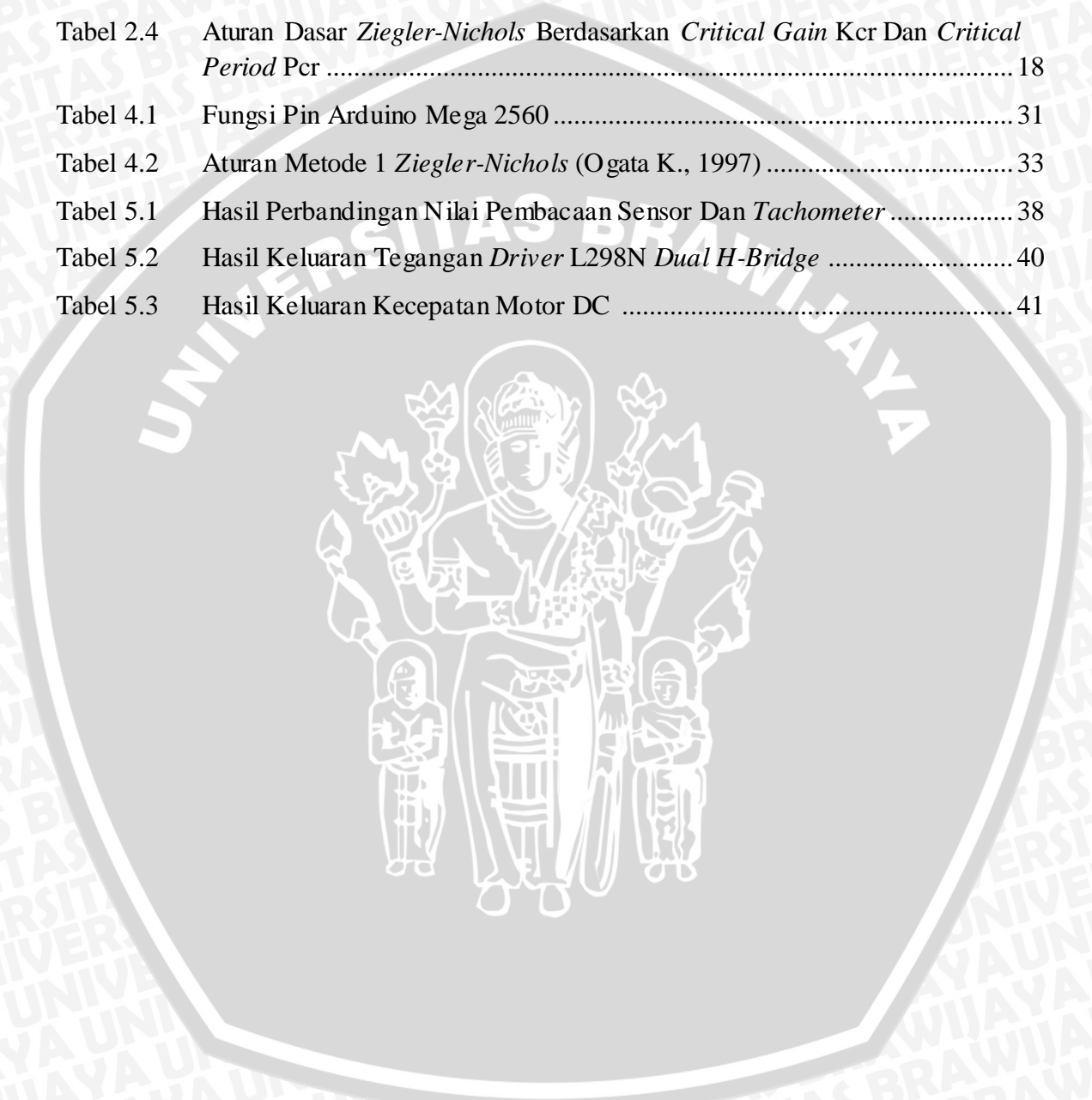
Tabel 4.1 Fungsi Pin Arduino Mega 2560 ..... 31

Tabel 4.2 Aturan Metode 1 *Ziegler-Nichols* (Ogata K., 1997) ..... 33

Tabel 5.1 Hasil Perbandingan Nilai Pembacaan Sensor Dan *Tachometer* ..... 38

Tabel 5.2 Hasil Keluaran Tegangan *Driver L298N Dual H-Bridge* ..... 40

Tabel 5.3 Hasil Keluaran Kecepatan Motor DC ..... 41



## BAB I PENDAHULUAN

### Latar Belakang

Setiap kali berada di sebuah lokasi proyek, ada sebuah struktur kokoh yang terbuat dari baja yang berdiri tinggi menjulang. Itulah *Tower Crane*. *Tower Crane* berfungsi sebagai alat angkat bahan - bahan dan material seperti mesin-mesin konstruksi, beton, besi, bekisting dan lain sebagainya. Karena besar dan fungsinya inilah *Tower Crane* merupakan alat berat dalam pekerjaan konstruksi. Cara kerja *Tower Crane* adalah dengan memindahkan barang bangunan secara vertikal dan horizontal. *Tower Crane* banyak digunakan untuk proyek-proyek pembangunan gedung-gedung bertingkat dan juga jembatan. Ukuran *Tower Crane* sangat beragam tergantung dari besarnya suatu bangunan atau jembatan yang akan dibangun. Pengendalian pada *Tower Crane* umumnya dikerjakan oleh tenaga kerja manusia. Tenaga kerja manusia yang digunakan juga harus memiliki kemampuan khusus dan terlatih agar dapat mengendalikan *Tower Crane* secara aman. Hal ini sangatlah tidak efisien karena berat barang yang akan dipindahkan sangatlah beragam. Selain itu Tenaga manusia juga memiliki batas tertentu dan dapat melakukan kesalahan kesalahan tertentu yang dapat membahayakan keselamatan pekerja.

Solusi untuk mengatasi masalah tersebut adalah dengan membuat *Tower Crane* otomatis untuk membantu pekerjaan manusia. Penelitian skripsi ini adalah untuk mendesain dan membangun *Tower Crane* yang kecepatannya dapat dikontrol sampai dengan *setpoint* yang diinginkan. Dengan mengatur kecepatan motor *Tower Crane* tersebut, diharapkan dapat memindahkan barang dengan waktu yang sama meskipun berat barang yang dipindahkan berbeda-beda. Serta untuk menghindari terjadinya kecelakaan kerja.

Sistem kontrol yang digunakan pada laporan pengembangan ini adalah Proporsional Integral Diferensial (PID). PID adalah kontroler yang merupakan gabungan dari kontroler proporsional, kontroler integral dan kontroler diferensial. 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 performa sistem yang didapatkan menjadi

stabil, reaksi sistem yang didapatkan menjadi lebih cepat, menghilangkan *offset* dan menghasilkan perubahan awal yang besar. Dengan menggunakan kontroler PID maka osilasi pada motor diharapkan dapat berkurang sehingga hasil sama seperti yang diinginkan, dan keunggulan dari laporan pengembangan ini adalah kecepatan putaran motor pada *Tower Crane* tersebut dapat dipertahankan sesuai dengan yang diinginkan.

## 1.2 Rumusan Masalah

Dari masalah yang telah disebutkan di atas, maka dalam skripsi ini rumusan masalah ditekankan pada:

1. Bagaimana merancang dan membuat sistem pengaturan kecepatan motor DC pada *Tower Crane* menggunakan kontroler PID ?
2. Bagaimana merancang *hardware* dan *software* sistem pengaturan kecepatan motor DC pada *Tower Crane* menggunakan kontroler PID ?

## 1.3 Batasan Masalah

Dalam perancangan untuk skripsi ini permasalahan dibatasi oleh hal-hal sebagai berikut :

1. Sistem ini merupakan model miniatur *Tower Crane* dengan desain sendiri.
2. Sensor yang digunakan adalah *rotary encoder* Autonic E40H8 500-6-L-5.
3. Digunakan 1 motor jenis motor DC *brushed*.
4. Pengereman yang dilakukan adalah pengaturan PWM pada motor DC.
5. Pembahasan ditekankan pada penggunaan kontroler PID pada sistem, untuk masalah elektrik dan sistematis tidak dibahas secara mendalam.
6. Berat maksimal beban material adalah kurang dari 200 gram.
7. *Setpoint* yang ditentukan sebesar 400 RPM.
8. Tinggi tiang *Tower Crane* 108 cm dan panjang jib *Tower Crane* 80 cm

## 1.4 Tujuan

Tujuan dari skripsi ini adalah untuk merancang sistem yang dapat mempertahankan kecepatan putaran motor DC pada *Tower Crane* saat terjadi gangguan berupa perubahan berat beban material dengan tetap mempertahankan keluaran sistem sesuai dengan *setpoint* yang diinginkan dengan menggunakan kontroler PID.

## 1.5 Sistematika Penulisan

Adapun sistematika dari penulisan skripsi ini adalah:

### **BAB I      Pendahuluan**

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

### **BAB II     Tinjauan Pustaka**

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

### **BAB III    Metodologi Penelitian**

Membahas metode penelitian dan perencanaan alat.

### **BAB IV     Perancangan Sistem**

Perancangan alat yang meliputi spesifikasi, perencanaan blok diagram, prinsip kerja dan pembuatan alat.

### **BAB V      Pengujian Alat**

Membahas hasil pengujian sistem dan analisa data secara keseluruhan terhadap alat yang telah direalisasikan.

### **BAB VI     Kesimpulan dan Saran**

Menjelaskan tentang pengambilan kesimpulan sesuai dengan hasil perancangan dan pengujian alat serta saran-saran yang diperlukan untuk melakukan pengembangan aplikasi selanjutnya.

UNIVERSITAS BRAWIJAYA





## BAB II

### TINJAUAN PUSTAKA

Untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan dari sistem yang dibuat, maka perlu adanya penjelasan dan uraian mengenai teori penunjang yang digunakan dalam penulisan skripsi ini. Teori penunjang yang akan dibahas dalam bab ini adalah:

- *Tower Crane*
- *Arduino Mega 2560*
- *Motor Direct Current (DC)*
- *Pulse width modulation (PWM)*
- *Sensor Rotary encoder*
- *Sensor Optocoupler*
- *Kontroler Proporsional Integral Diferensial (PID)*
- *Driver Motor DC L298n*

#### 2.1 Pengertian *Tower Crane*

*Tower Crane* adalah alat pengangkut yang digunakan untuk mengangkut mesin dan bahan-bahan bangunan. *Tower Crane* memiliki mekanisme mengangkat, berputar, dan trolis. *Tower Crane* umumnya digunakan pada pembangunan gedung-gedung tinggi dan jembatan. *Tower Crane* dikendalikan oleh tenaga kerja manusia dan memiliki kokpit yang berisi panel-panel untuk menggerakkan *Tower Crane*. Bentuk *Tower Crane* ditunjukkan pada Gambar 2.1



**Gambar 2.1 Tower Crane**Sumber: <http://id.wikipedia.org/wiki/TowerCrane>**2.1.1 Kecepatan Tower Crane**

Kecepatan *Tower Crane* pada umumnya berbeda sesuai dengan kapasitas berat dari *Tower Crane*. Kecepatan tiap mekanisme juga berbeda-beda. Tabel 2.1 adalah spesifikasi kecepatan *Tower Crane* tipe JL4210A produksi Xiangtan Jiangu Import and Export Co.,Ltd.

**Tabel 2.1 Kecepatan Dan Kapasitas Berat Tower Crane**

Kecepatan (m/min)	Kapasitas berat (t)
7.5	4
35	2
70	1

Sedangkan untuk mekanisme troli pada *Tower Crane* tipe JL4210A memiliki kecepatan 0-40 m/min dan kecepatan mekanisme rotasinya adalah 0-0.7 r/min.

**2.1.2 Kapasitas berat Tower Crane**

*Tower Crane* memiliki batas maksimal untuk mengangkut suatu beban. Kapasitas berat *Tower Crane* juga bergantung pada jarak horizontal *jib*. *Jib* adalah bagian horizontal pada puncak *Tower Crane*. Sehingga batas maksimal kapasitas berat pada *Tower Crane* juga bergantung pada jarak *jib*. Semakin dekat posisi beban yang diangkat dengan tiang *Tower Crane*, semakin besar beban yang dapat diangkat dengan aman dan sebaliknya, semakin jauh posisi beban yang diangkat dengan tiang *Tower Crane*, semakin kecil beban yang dapat diangkat dengan aman. *Tower Crane* juga dilengkapi *Counter Weight* yang berfungsi sebagai beban penyeimbang agar *Tower Crane* lebih stabil dan tidak menahan beban yang tinggi. Tabel 2.2 adalah spesifikasi kapasitas berat *Tower Crane* tipe JL4210A produksi Xiangtan Jiangu Import and Export Co.,Ltd.

**Tabel 2.2 Spesifikasi Tower Crane Tipe JL4210A**

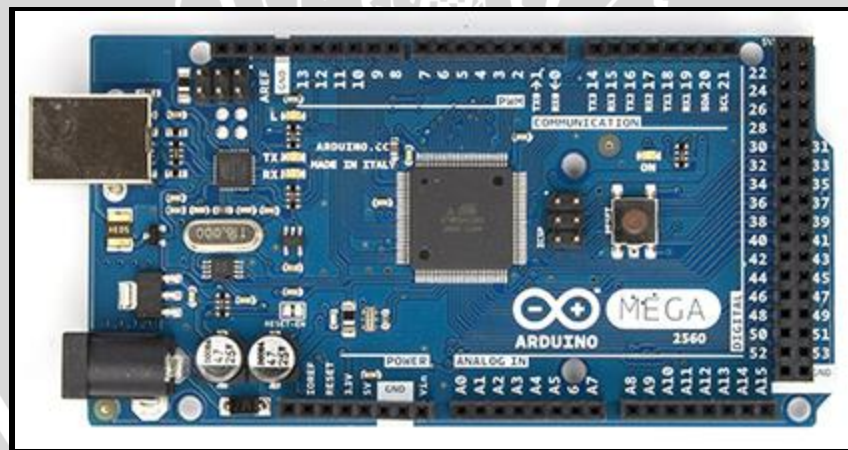
Berat maksimum (t)	Jarak <i>jib</i> terhadap tiang TC (m)
4	13.2
1.6	37

### 2.1.3 Alat Penggerak *Tower Crane*

Pada umumnya *Tower Crane* digerakkan oleh sebuah motor pada setiap mekanismenya. Pada bagian tengah *Tower Crane* terdapat engsel berputar yang dilengkapi motor sehingga membuat *Tower Crane* dapat berubah arah hingga  $360^\circ$ . Pada bagian horizontal *jib* terdapat *fall* baja yang dilengkapi dengan tali baja yang digerakkan oleh motor sehingga dapat mengangkat beban secara vertikal, dan juga terdapat *travel trolley* yang digerakkan oleh motor sehingga dapat memindahkan barang secara horizontal.

## 2.2 Arduino Mega 2560

Arduino mega 2560 adalah *board* arduino Mega versi terbaru, yang merupakan perbaikan dari *board* terdahulunya yaitu arduino Mega 1280. Arduino mega 2560 memiliki 54 digital *input / output* ( dengan 14 pin yang dapat digunakan sebagai *output* PWM (*Pulse width modulation*)), 16 *input* analog, 4 UART (*Universal Asynchronous Receiver / Transmitter*), kristal 16 MHz osilator, koneksi USB, jack listrik, *header* ICSP (*in-circuit serial programming*), dan tombol reset. Bentuk Arduino mega 2560 ditunjukkan pada Gambar 2.2.



Gambar 2.2 Arduino Mega 2560 Tampak Atas  
Sumber: Arduino Mega 2560 Datasheet

### 2.2.1 Daya

Arduino mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya akan dipilih secara otomatis. Eksternal (non-USB) daya dapat berasal baik dari adaptor AC-DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan sebuah 2.1 mm konektor pusat-positif ke colokan listrik *board*.

Modul dapat beroperasi pada pasokan eksternal 6 sampai 20 volt. Jika disertakan dengan kurang dari 7V, pin 5V dapat menyediakan kurang dari lima volt dan modul mungkin tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak modul. Kisaran yang disarankan adalah 7 sampai 12 volt.

### 2.2.2 Memori

Atmega 2560 memiliki 256 KB memori *flash* untuk menyimpan kode dengan 8 KB digunakan untuk *bootloader*, 8 KB SRAM (*Static Random Access Memory*) dan 4 KB EEPROM (*Electrically Erasable Programmable Read-Only Memory*).

### 2.2.3 Input Dan Output

Masing-masing dari 54 digital pin pada Mega dapat digunakan sebagai *input* atau *output*, menggunakan *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Dengan 14 pin yang dapat digunakan sebagai *output* PWM (*Pulse width modulation*) menyediakan 8-bit PWM *output* dengan fungsi *analogWrite()*.

### 2.2.4 Komunikasi

Arduino Mega2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega2560 ini menyediakan empat UART *hardware* untuk komunikasi serial. LED (*Light-Emitting Diode*) akan berkedip ketika data sedang dikirim melalui ATmega8U2/ATmega16U2 koneksi Chip dan USB ke komputer.

## 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.

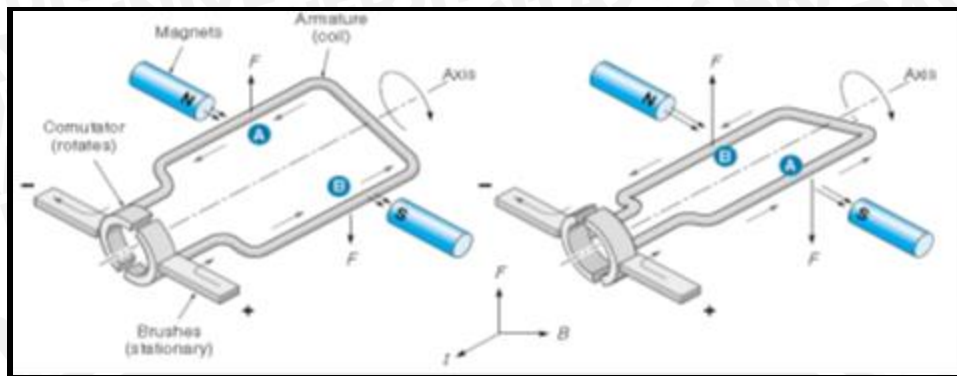
Kaidah tangan kiri untuk motor menunjukkan arah arus yang mengalir didalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah medan putar. Adapun besarnya gaya yang bekerja pada konduktor tersebut dapat dirumuskan dalam persamaan (2.1). Cara kerja motor DC ditunjukkan pada Gambar 2.3.

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

$B$  = kerapatan fluks magnet (*weber*)

$L$  = panjang konduktor (*meter*)

$I$  = arus listrik (*ampere*)



**Gambar 2.3** Cara Kerja Motor DC

Sumber : Kilian, 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.

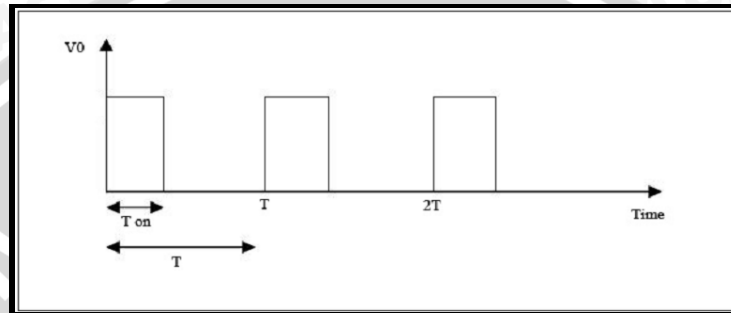
Pada kawat yang berada di kanan arus mengalir dari depan ke belakang. Pada kawat yang berada di bagian kiri, arus mengalir dari belakang ke depan kawat a dan b secara bergantian berada di kiri dan kanan. Karena itu arah arus di a dan arah arus di b selalu bersifat bolak-balik. Pembalikan arah arus itu 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 bekerja sama kuatnya, sehingga terdapat kopel yang bekerja pada kawat sehingga lilitan jangkar dapat berputar. Setelah berputar  $180^\circ$  arah arus berbalik, pada saat itu penghantar a dan b bertukar tempat. Akibatnya arah gerak putaran tidak berubah.

## 2.4 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. *Duty cycle* adalah rasio atau perbandingan lama waktu suatu sinyal berada dalam kondisi *high* dengan lama waktu suatu sinyal tersebut dalam kondisi (*high+low*).

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM secara umum dapat dilihat dalam Gambar 2.4.



Gambar 2.4 Sinyal PWM Secara Umum

Sumber: [www.electronics-scheme.com](http://www.electronics-scheme.com)

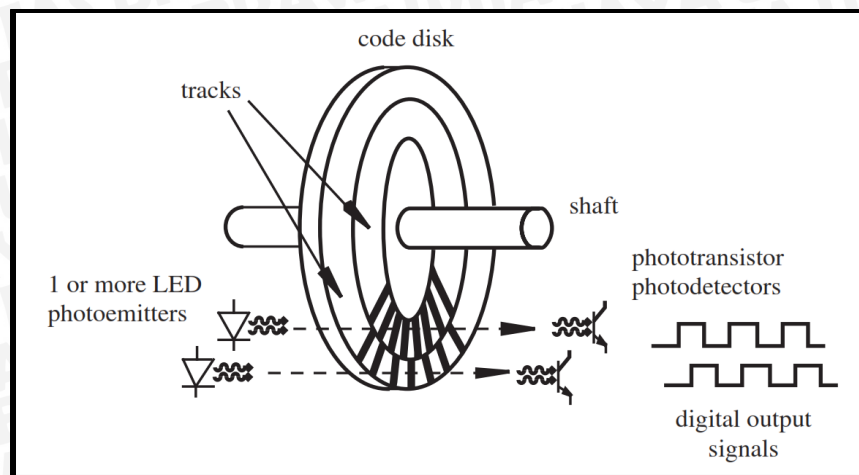
$$\text{Duty cycle} = \frac{T_{\text{on}}}{T} \times 100\% \dots (\%) \quad (2.2)$$

Dengan :

$$\begin{aligned} T_{\text{on}} &= \text{Periode logika tinggi} \\ T &= \text{Periode keseluruhan} \end{aligned}$$

## 2.5 Sensor Rotary Encoder

Sensor putaran (*rotary encoder*) digunakan untuk mendeteksi banyak putaran pada motor DC. Salah satu *rotary encoder* yang sering digunakan adalah jenis optik. Sensor ini terdiri dari *Light Emitting Diode* (LED) inframerah dan *phototransistor*. Secara prinsip sensor mendeteksi halangan diantara LED inframerah dan *phototransistor*. Apabila cahaya dari LED tidak terhalang maka cahaya dari LED akan mengenai *phototransistor* sehingga *phototransistor* akan aktif. Apabila cahaya LED terhalang maka *phototransistor* tidak aktif. Diagram *rotary encoder* ditunjukkan dalam Gambar 2.5.



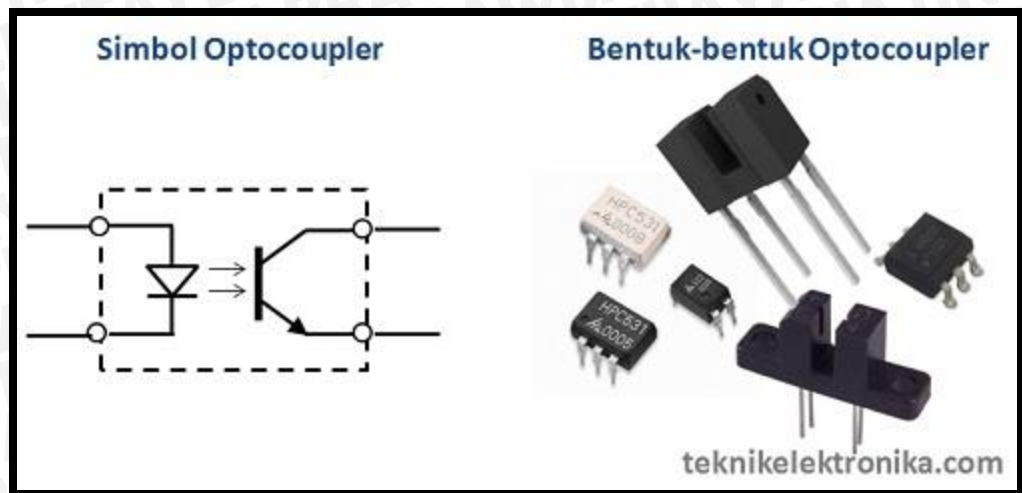
**Gambar 2.5. Diagram Rotary Encoder**

Sumber : Alciatore dan Hinstead (2012 : 384)

*Incremental rotary encoder* atau kadang-kadang disebut *rotary encoder* relatif dalam hal desain lebih sederhana daripada *absolute rotary encoder*. *incremental rotary encoder* yang memiliki keluaran sinyal *quadrature* terdiri dari dua LED dan dua *phototransistor* dan memiliki keluaran yang disebut keluaran A dan B. Saat porosnya diputar, *rotary encoder* mengeluarkan pulsa yang frekuensinya sebanding dengan kecepatan rotasi poros. Pola keluaran pada saluran A dan B ditunjukkan dalam Gambar 2.5. Dengan menghitung jumlah pulsa dan mengetahui resolusi *rotary encoder*, jumlah putaran dapat diukur. Keluaran A dan B digunakan untuk mengetahui arah rotasi dengan melihat saluran mana yang mendahului.

## 2.6 *Optocoupler*

*Optocoupler* adalah komponen elektronika yang berfungsi sebagai penghubung berdasarkan cahaya optik. Pada dasarnya *Optocoupler* terdiri dari 2 bagian utama yaitu *transmitter* yang berfungsi sebagai pengirim cahaya optik dan *receiver* yang berfungsi sebagai pendeteksi sumber cahaya. Masing-masing bagian *Optocoupler* (*transmitter* dan *receiver*) tidak memiliki hubungan konduktif rangkaian secara langsung tetapi dibuat sedemikian rupa dalam satu kemasan komponen. Bentuk *Optocoupler* ditunjukkan pada Gambar 2.6.



Gambar 2.6. Optocoupler

Sumber : <http://teknikelektronika.com>

## 2.7 Kontroler

Sistem pengendalian dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengendalian adalah harus stabil. Di samping 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 merespon *input* dan peredamannya terhadap adanya lonjakan (*overshoot*).

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

Dalam sistem pengendalian terdapat 2 macam *loop*:

1. Pengendalian dengan *loop* terbuka



Sistem kontrol *loop* terbuka adalah sistem kontrol yang keluarannya tidak berpengaruh pada aksi pengontrolan. Jadi pada sistem kontrol *loop* terbuka, keluaran tidak diukur atau diumpam balik untuk dibandingkan dengan masukan.

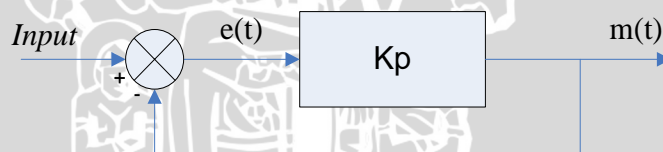
## 2. Pengendalian dengan *loop* tertutup

Sistem kontrol *loop* tertutup adalah sistem kontrol yang keluarannya mempunyai pengaruh langsung pada aksi pengontrolan. Disebut juga sistem kontrol yang menggunakan umpan balik untuk memperkecil kesalahan sistem.

### 2.7.1 Kontroler Proporsional

Kontroler proporsional memiliki keluaran yang sebanding/proporsional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Secara lebih sederhana dapat dikatakan, bahwa keluaran kontroler proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan pada sinyal masukan akan segera menyebabkan sistem secara langsung mengubah keluarannya sebesar konstanta pengalinya.

Pada Gambar 2.7 menunjukkan blok diagram yang menggambarkan hubungan antara *input* (besaran referensi yang diinginkan), besaran aktual dengan besaran keluaran kontroler Proporsional, dan besaran kesalahan (*error*). Sinyal kesalahan (*error*) merupakan selisih antara besaran *setting* dengan besaran aktualnya.



Gambar 2.7 Diagram Blok Kontroler Proporsional

Sumber: Ogata,1995: 157

Pada pengendali proporsional hubungan antara keluaran kontroler  $m(t)$  dan sinyal kesalahan  $e(t)$  adalah

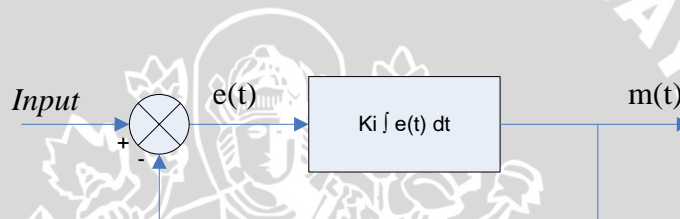
$$m(t) = K_p e(t) \quad (2.3)$$

dengan  $K_p$  adalah penguatan proporsional. Keluaran  $m(t)$  hanya tergantung pada  $K_p$  dan *error*, semakin besar *error* maka semakin besar koreksi yang dilakukan. Penambahan  $K_p$  akan menaikkan penguatan sistem sehingga dapat digunakan untuk memperbesar kecepatan respon dan mengurangi kesalahan keadaan mantap.

## 2.7.2 Kontroler Integral

Kontroler integral berfungsi mengurangi kesalahan keadaan mantap yang dihasilkan pada kontroler proporsional sebelumnya. Kalau sebuah plant tidak memiliki unsur integrator ( $1/s$ ), kontroler proporsional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan mantapnya nol.

Kontroler integral memiliki karakteristik seperti halnya sebuah integral. Keluaran kontroler sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran kontroler ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Gambar 2.8 menunjukkan blok diagram kontroler integral.



Gambar 2.8 Diagram Blok Kontroler Integral  
Sumber: Ogata, 1995: 158

Nilai keluaran kontroler  $m(t)$  sebanding dengan integral sinyal kesalahan  $e(t)$ ,

Sehingga

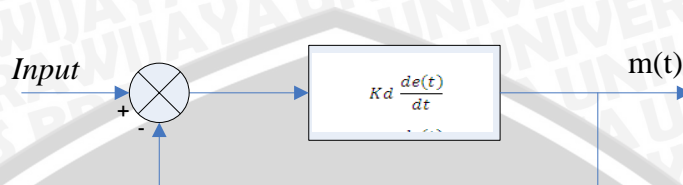
$$\frac{dm(t)}{dt} = Ki \cdot e(t) \quad (2.4)$$

$$m(i) = Ki \int_0^t e(t)dt \quad (2.5)$$

dengan  $K_i$  adalah konstanta integral. Jika sinyal kesalahan  $e(t)=0$ , maka laju perubahan sinyal kendali integral  $\frac{dm(t)}{dt} = 0$  atau sinyal keluaran kendali akan tetap berada pada nilai yang dicapai sebelumnya. Aksi control integral digunakan untuk menghilangkan kesalahan posisi dalam keadaan mantap (*error steady state*) tanpa memperhitungkan kecepatan respon.

### 2.7.3 Kontroler Diferensial

Kontroler diferensial memiliki sifat seperti halnya suatu operasi *derivatif*. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat. Gambar 2.9 berikut menunjukkan blok diagram pada kontroler diferensial.



Gambar 2.9 Blok Diagram Kontroler Diferensial

Sumber: Ogata, 1995: 177

Nilai keluaran kontroler  $m(t)$  sebanding laju sinyal kesalahan  $\frac{de(t)}{dt}$ . Hubungan ini dapat ditulis sebagai:

$$m(t) = Kd \frac{de(t)}{dt} \quad (2.6)$$

Kontroler diferensial akan memberikan sinyal kendali keluaran  $m(t) = 0$ , untuk sinyal kesalahan  $e(t)$  yang konstan sehingga kontroler diferensial tidak mempengaruhi keadaan mantap. Kontroler diferensial digunakan untuk memperbaiki atau mempercepat respon transien sebuah sistem serta dapat meredam osilasi.

Berdasarkan karakteristik kontroler tersebut, kontroler diferensial umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja kontroler diferensial hanyalah efek dari lingkup yang sempit, yaitu pada periode peralihan. Oleh sebab itu kontroler differensial tidak bisa digunakan tanpa ada kontroler lain.

### 2.7.4 Kontroler Proporsional Integral Diferensial (PID)

Setiap kekurangan dan kelebihan dari masing-masing kontroler P,I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler proporsional integral differensial (PID). Elemen-elemen kontroler P,I dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem,

menghilangkan *offset* dan menghasilkan perubahan awal yang besar (Gunterus, 1994, 8-10).

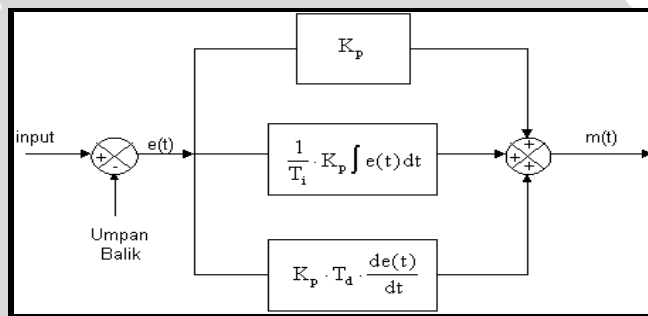
Persamaan kontroler PID ini dapat dinyatakan dalam persamaan berikut:

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2.7)$$

Dalam transformasi Laplace dinyatakan sebagai berikut:

$$\frac{M(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \quad (2.8)$$

Jenis kontroler ini digunakan untuk memperbaiki kecepatan respon, mencegah terjadinya kesalahan keadaan mantap serta mempertahankan kestabilan. Kontroler PID memiliki diagram kendali seperti yang ditunjukkan dalam Gambar 2.10.



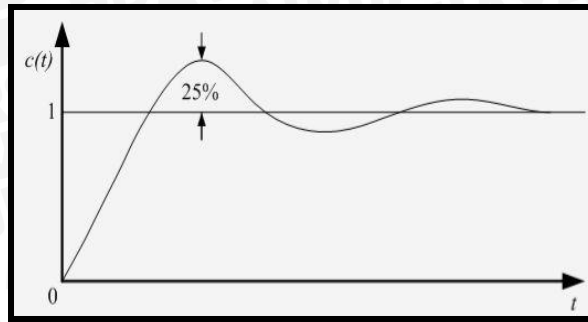
Gambar 2.10 Diagram Blok Kontroler PID  
Sumber: Ogata K., 1997

## 2.8 Metode Perancangan Kontroler Proporsional Integral Diferensial (PID)

### 2.8.1 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).

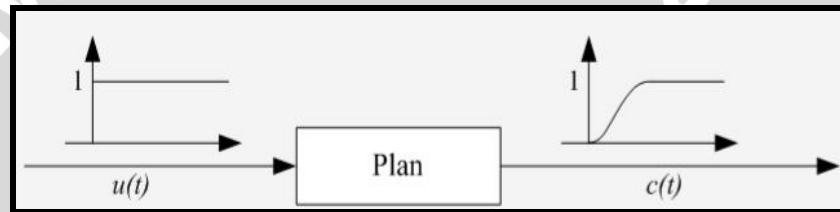
Terdapat dua metode yang disebut dengan aturan penalaan *Ziegler-Nichols*, pada kedua metode tersebut memiliki tujuan yang sama yaitu untuk mencapai 25% *maximum overshoot* pada respon *unit step*, seperti ditunjukkan dalam Gambar 2.11.



Gambar 2.11 Kurva Respon Unit Step yang Menunjukkan 25% Maximum Overshoot  
 Sumber: Ogata, K., 1997

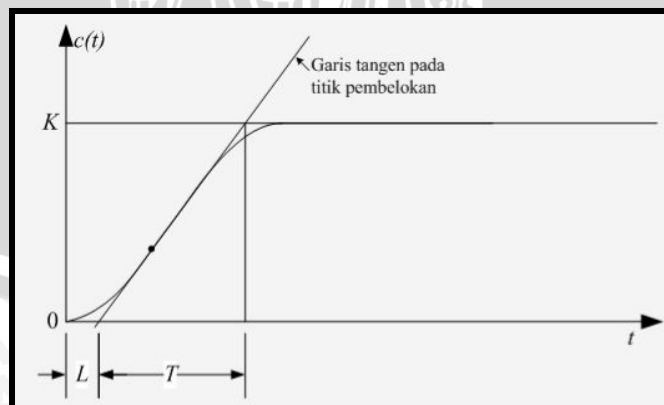
a). Metode Pertama

Metode pertama atau sering disebut metode kurva reaksi, respon dari plan dapat diperoleh secara eksperimental dengan masukan berupa *unit step*, seperti yang ditunjukkan dalam Gambar 2.12.



Gambar 2.12 Respon Plant Terhadap Masukan Berupa Unit Step  
 Sumber: 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 2.13, jika respon tidak memberikan bentuk kurva S, maka metode ini tidak berlaku. (Ogata, K., 1997).



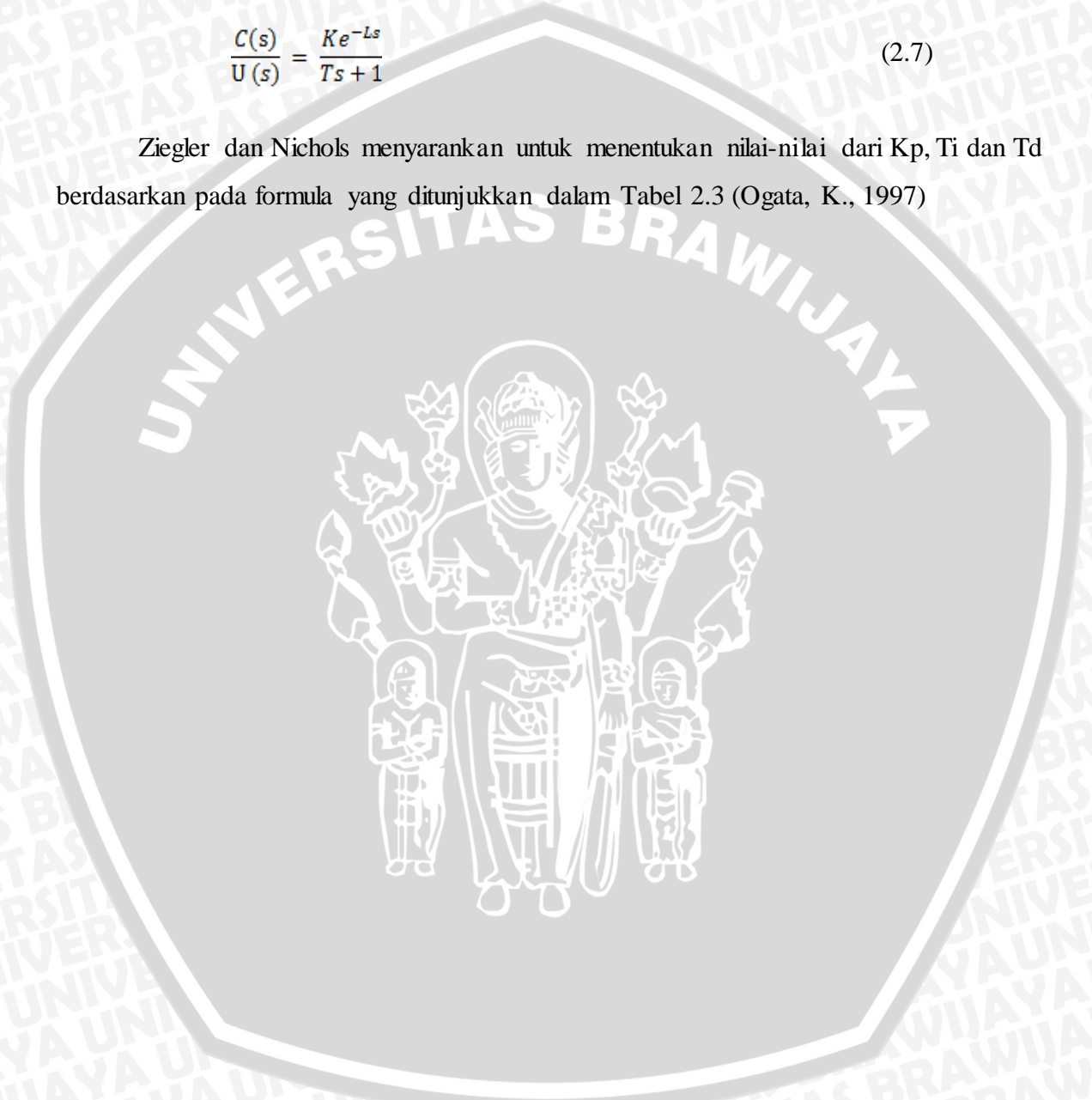
Gambar 2.13 Kurva Respon Yang Berbentuk S  
 Sumber: Ogata, K. 1997



Kurva berbentuk S tersebut dapat dikarakteristikkan 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 2.13. Fungsi alih  $C(s)/U(s)$  dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut:

$$\frac{C(s)}{U(s)} = \frac{K e^{-Ls}}{Ts + 1} \quad (2.7)$$

Ziegler dan Nichols menyarankan untuk menentukan nilai-nilai dari  $K_p$ ,  $T_i$  dan  $T_d$  berdasarkan pada formula yang ditunjukkan dalam Tabel 2.3 (Ogata, K., 1997)



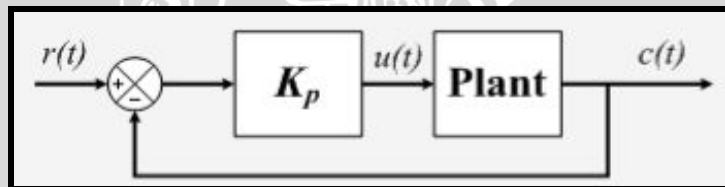
Tabel 2.3 Aturan Penalaan Ziegler-Nichols Berdasarkan Respon Unit Step Dari Plan

Tipe Kontroler	$K_p$	$T_i$	$T_d$
P	$\frac{T}{L}$	$\infty$	0
PI	$0,9 \frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{T}{L}$	$2L$	$0,5 L$

Sumber: Ogata, K. 1997

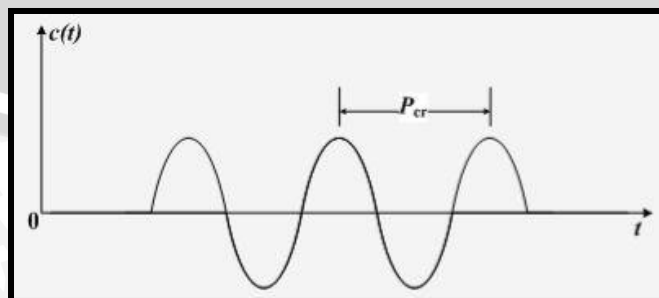
b). Metode Kedua

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 2.14 dan untuk osilasi berkesinambungan dengan periode  $P_{cr}$  dapat dilihat dalam Gambar 2.15. Ziegler dan Nichols menyarankan penyetelan nilai parameter  $K_p, T_i, T_d$  dan berdasarkan rumus yang diperlihatkan dalam Tabel 2.4. (Ogata, K., 1997).



Gambar 2.14 Sistem Loop Tertutup dengan Kontroler Proporsional

Sumber: Ogata, K., 1997



Gambar 2.15 Osilasi Berkesinambungan dengan Periode  $P_{cr}$

Sumber : Ogata, K., 1997

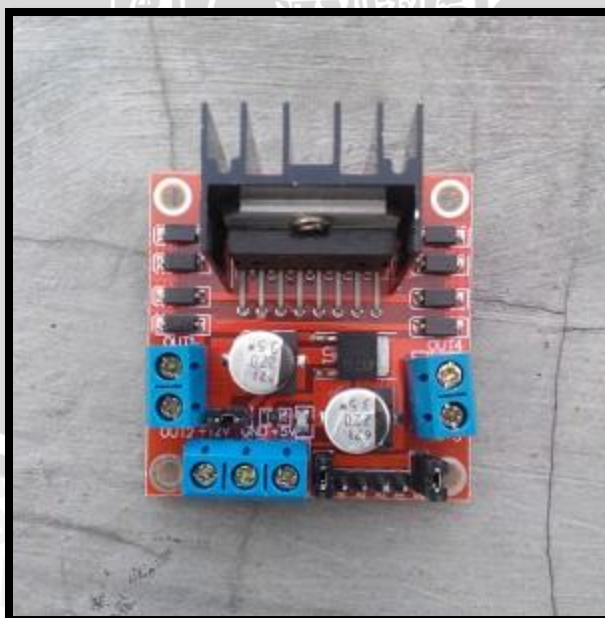
Tabel 2.4 Aturan Dasar Ziegler-Nichols Berdasarkan *Critical Gain Kcr* Dan *Critical Period Pcr*

Tipe Kontroler	$K_p$	$T_i$	$T_d$
P	0.5 Kcr	$\infty$	0
PI	0.45 Kcr	$\frac{1}{1,2} Pcr$	0
PID	0.60 Kcr	0.5 Pcr	0.125 Pcr

Sumber: Ogata, K., 1997

### 2.9 Driver motor DC L298n Dual H-Bridge

L298 adalah jenis IC *driver* motor yang dapat mengendalikan arah putaran dan kecepatan motor DC ataupun Motor *stepper*. Mampu mengeluarkan *output* tegangan untuk Motor dc dan motor *stepper* sebesar 50 volt. IC l298 terdiri dari transistor-transistor logik (TTL) dengan gerbang nand yang memudahkan dalam menentukan arah putaran suatu motor dc dan motor *stepper*. Dapat mengendalikan 2 untuk motor dc namun pada hanya dapat mengendalikan 1 motor *stepper*. Penggunaannya paling sering untuk robot *line follower*. Bentuknya yang kecil memungkinkan dapat meminimalkan pembuatan robot *line follower*. Bentuk L298N *Dual H-Bridge* ditunjukkan pada Gambar 2.16.



Gambar 2.16 Module *driver* L298N *Dual H-Bridge*

Sumber <http://jualarduinomurah.com/l298n-module/>



UNIVERSITAS BRAWIJAYA



## BAB III

### METODOLOGI PENELITIAN

Untuk menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan maka diperlukan langkah-langkah untuk menyelesaikan masalah tersebut. Metode yang digunakan dapat diuraikan sebagai berikut:

#### 3.1 Studi Literatur

Studi literatur digunakan untuk mempelajari teori dan ilmu pengetahuan yang berhubungan dengan permasalahan yang akan diteliti. Sumber literatur diperoleh dari buku, jurnal, serta studi terhadap penelitian terdahulu yang terkait dengan topik utama dalam penelitian.

#### 3.2 Penentuan Spesifikasi Alat

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

- Desain miniatur *Tower Crane* berbahan dasar *stainless steel* dan *alluminium*.
- Desain alat berbentuk tiang dengan ukuran tinggi 100 cm. Pada ujung atasnya terdapat *jib Tower Crane* dengan panjang 80 cm, bagian depan 50 cm dan belakang 30 cm dari sumbu tiang.
- Menggunakan satu buah motor DC 12 volt sebagai penggerak.
- Menggunakan satu buah *rotary encoder* yang dikopel dengan motor DC sebagai sensor kecepatan
- Menggunakan satu buah *optocoupler* yang diletakkan di ujung atas *Tower Crane* sebagai sensor jarak.
- Mikrokontroler yang digunakan Arduino Mega 2560.
- Kontroler yang digunakan adalah kontroler PID.

### 3.3 Perancangan Sistem

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan agar sistem dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan.

Perancangan sistem yang dilakukan meliputi:

1. Penentuan sensor yang akan digunakan untuk mendeteksi kecepatan putaran motor DC dan posisi.
2. Penentuan motor DC yang digunakan sebagai *Tower Crane*
3. Penentuan spesifikasi sistem yang akan dibuat, meliputi :
  - a. Penentuan deskripsi kerja sistem secara keseluruhan.
  - b. Kontroler PID dan rangkaian elektronik pendukung.
  - c. Perancangan program pada Mikrokontroler sebagai pusat pengontrol dengan kontroler PID.

### 3.4 Realisasi Pembuatan Sistem

#### 3.4.1 Perangkat Keras

Komponen utama perangkat keras adalah desain miniatur *Tower Crane* yang berbahan dasar *stainless steel*, motor DC, pendeteksi kecepatan menggunakan *rotary encoder autonic E40H8 500-6-L-5*, pendeteksi posisi puncak *Tower Crane* menggunakan 1 buah *optocoupler*, *driver* motor L298n dan Mikrokontroler Arduino Mega 2560.

#### 3.4.2 Perangkat Lunak

Pembuatan perangkat lunak (*software*) dilakukan dengan pembuatan *flowchart* terlebih dahulu, kemudian dilakukan perancangan program pada *software* Arduino Mega 2560 sebagai pusat pengontrol dengan kontrol PID.

### 3.5 Pengujian Alat

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

1. Menguji pada tiap-tiap blok rangkaian.
2. Menggabungkan beberapa blok rangkaian menjadi sebuah sistem.
3. Mengadakan pengujian sistem secara keseluruhan.
4. Mengevaluasi hasil pengujian sistem secara keseluruhan

### 3.6 Pengambilan Kesimpulan dan Saran

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



UNIVERSITAS BRAWIJAYA



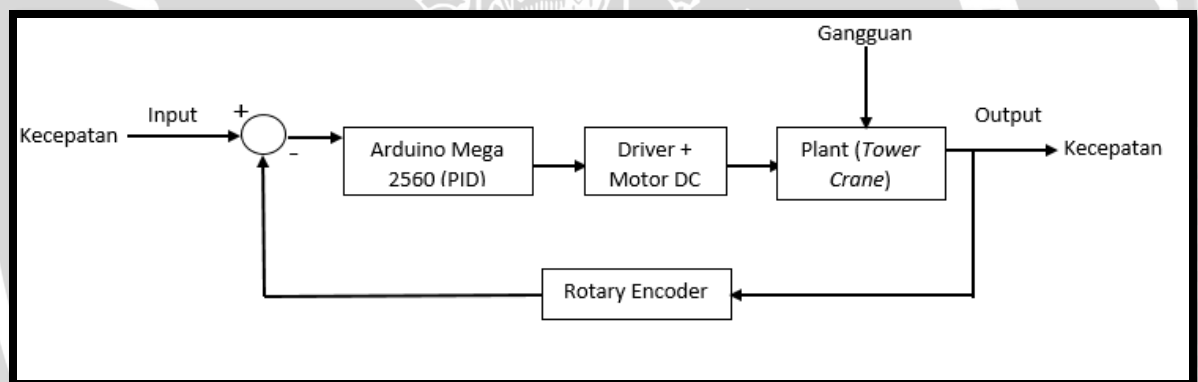
## BAB IV

### PERANCANGAN DAN PEMBUATAN ALAT

Perancangan dan pembuatan dalam skripsi ini bertujuan untuk merancang beberapa perangkat maupun alat secara keseluruhan. Perancangan perangkat tersebut meliputi perancangan perangkat keras maupun perancangan perangkat lunak. Sedangkan pembuatan bertujuan untuk menghasilkan semua perangkat pendukung maupun alat secara keseluruhan.

#### 4.1 Diagram Blok Sistem

Pada perancangan alat diperlukan perancangan blok diagram sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai dengan rencana. Blok diagram tersebut dapat dilihat dari Gambar 4.1 berikut.



Gambar 4.1 Blok Diagram Sistem

Keterangan dari blok diagram pada Gambar 4.1 adalah sebagai berikut :

- *Setpoint* sistem berupa kecepatan dengan satuan RPM, dan posisi yang menunjukkan bahwa beban berada di dasar atau ujung *tower crane*.
- Sensor kecepatan yang digunakan berupa *Rotary Encoder Autonic E40H8 500-6-L-5* yang berfungsi mendeteksi kecepatan motor pada saat motor bergerak dan akan memberikan masukan berupa pulsa ke Arduino Mega 2560.
- Sensor posisi yang digunakan berupa 1 buah sensor *optocoupler*.
- Pusat pengendalian sistem menggunakan Arduino Mega 2560 yang memberikan keluaran berupa *pulse width modulation (PWM)* kepada *driver motor*.

- Motor yang digunakan adalah motor DC yang bergerak secara linier dan berfungsi sebagai aktuator.
- *Driver* motor menggunakan modul L298N *Dual H-Bridge*.

#### 4.2 Prinsip Kerja Alat

Cara kerja alat adalah sebagai berikut :

- Menggunakan catu daya sebesar 12 volt untuk motor DC dan 5 volt untuk sensor *Rotary Encoder Autonic E40H8 500-6-L-5*.
- *Rotary Encoder Autonic E40H8 500-6-L-5* sebagai sensor kecepatan motor DC. Sensor *Rotary encoder* akan memberikan keluaran berupa pulsa yang berubah-ubah sesuai dengan perubahan kecepatan.
- 1 buah sensor *optocoupler* sebagai sensor posisi, yang berfungsi sebagai pendeteksi saat pengereman *tower crane*.
- Port komunikasi antara Arduino Mega 2560 dan PC menggunakan perantara kabel USB dengan kecepatan transfer data sebesar 9600 bps.
- Masukan sistem berasal dari *push button* (saklar). *Push button* berfungsi sebagai penentu gerak yang akan dituju sesuai dengan kebutuhan.
- Untuk menggerakkan *tower crane* ke posisi turun, *driver* motor harus berlogika rendah atau 0, keadaan itu dapat dirancang di dalam program. Saat *setpoint* ditentukan, maka motor DC yang telah dirancang akan bergerak linier dengan kecepatan sesuai *setpoint* yang ditentukan.
- Untuk menggerakkan *tower crane* ke posisi naik, *driver* motor harus berlogika tinggi atau 1, keadaan itu dapat dirancang di dalam program. Saat *setpoint* ditentukan, maka motor DC yang telah dirancang akan bergerak linier dengan kecepatan sesuai *setpoint* yang ditentukan.
- Untuk proses pengereman *tower crane* berdasarkan dari total pulsa keseluruhan proses naik atau turun. Pulsa yang terbaca pada keseluruhan proses adalah 12500. Sehingga pada saat proses sistem mencapai pulsa 10000 maka program akan mengatur PWM agar kecepatan motor DC berkurang hingga akhirnya berhenti total pada pulsa 12500.

#### 4.3 Perancangan Perangkat Keras

Berdasarkan diagram blok perancangan alat yang telah disusun, perangkat keras meliputi model miniatur *tower crane*, rangkaian catu daya, *driver* motor, motor DC, sensor

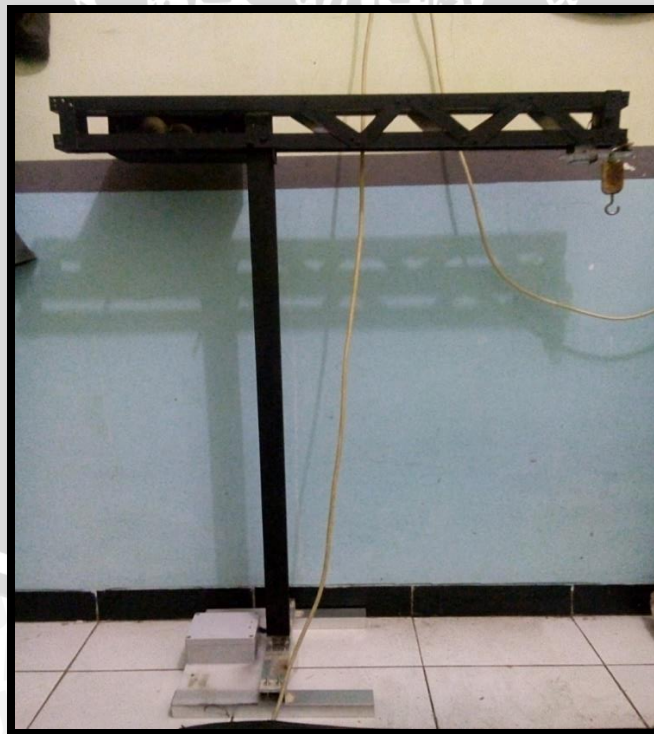
repository.ub.ac.id

*rotary encoder*, penentuan parameter penguatan kontroler, modul Arduino Mega 2560. Di bawah ini adalah penjelasan masing-masing rangkaian penyusun keseluruhan alat.

#### 4.3.1 Spesifikasi Alat

Spesifikasi model miniatur *tower crane* adalah sebagai berikut :

- Model miniatur *tower crane* yang dibuat adalah *tower crane* dengan desain sendiri.
- *Tower crane* berbentuk kerangka 1 tiang penyangga dengan ukuran tinggi 108 cm, panjang jib *tower crane* 80 cm, jarak antara *jib* dengan tiang sumbu 50 cm dan 30 cm.
- Penarik katrol *tower crane* menggunakan senar pancing nilon dengan tebal 0.4 mm
- Menggunakan satu buah motor DC sebagai penggerak.
- Sensor kecepatan dan jarak yang digunakan adalah *Rotary Encoder Autonic E40H8 500-6-L-5* yang dipasang sejajar dengan motor DC.
- Sensor posisi yang digunakan adalah 1 buah *optocoupler* yang terpasang pada ujung jib *tower crane*.
- Penggerak *tower crane* berupa motor DC dengan catu daya 12 volt.
- Model miniatur *tower crane* dapat dilihat dalam Gambar 4.2, 4.3, dan 4.4
- Penempatan sensor *optocoupler* dapat dilihat dalam Gambar 4.5.



Gambar 4.2 Rangka Model Miniatur





Gambar 4.3 Rotary Encoder Sejajar Dengan Motor DC



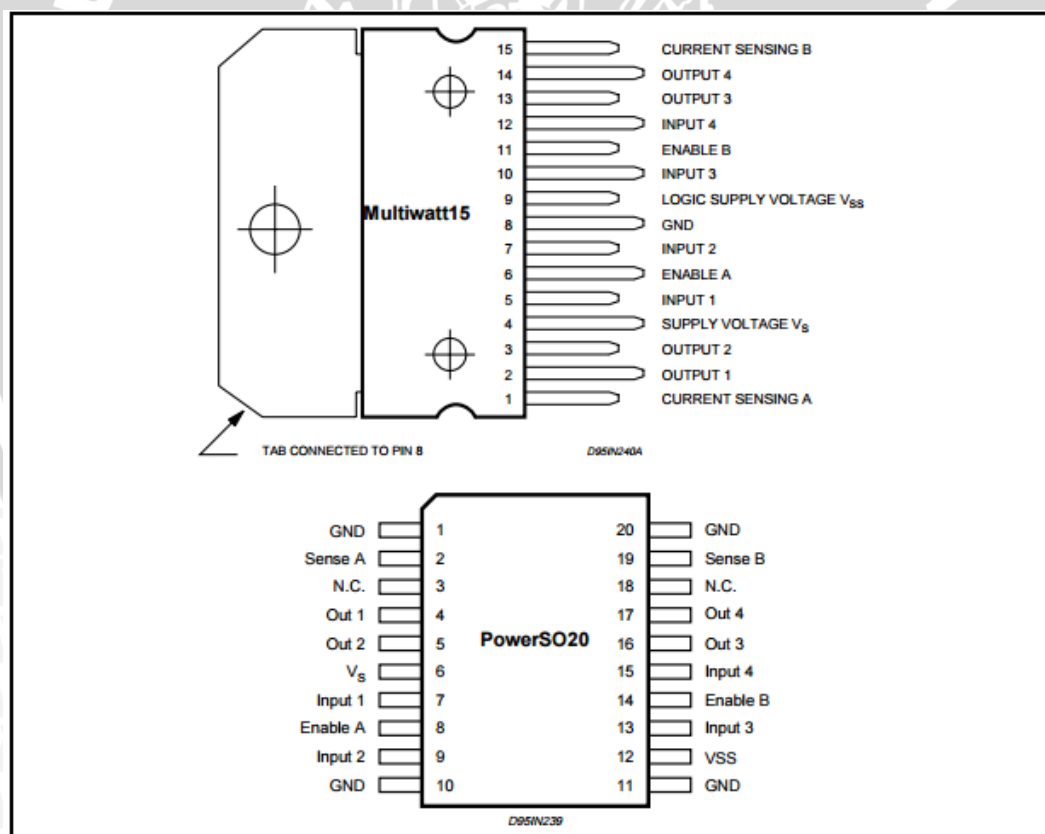
Gambar 4.4 Model Miniatur Tower Crane



Gambar 4.5 Penempatan Sensor *Optocoupler*

### 4.3.2 Perancangan Driver Motor DC

Modul pengendali motor DC yang digunakan adalah L298N *Dual H-Bridge*. Secara garis besar, fungsi modul pengendali motor ini adalah untuk mengendalikan arah dan kecepatan putaran motor DC sesuai instruksi kendali dari Arduino Mega 2560. Gambar koneksi modul pengendali motor DC L298N *Dual H-Bridge* ditunjukkan dalam Gambar 4.7

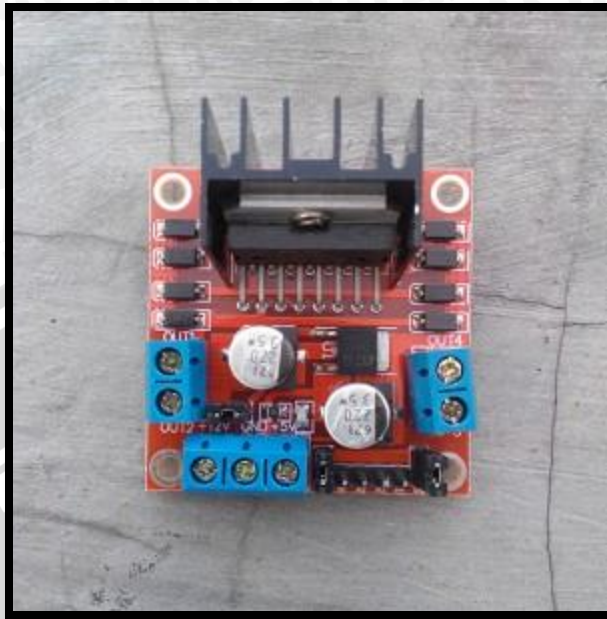


Gambar 4.6 Koneksi Modul L298N *Dual H-Bridge*

Sumber: Datasheet L298N *Dual H-Bridge*



Rangkaian *driver* ini sanggup bekerja dengan tegangan maksimal 40 *volt*, serta kapasitas arus maksimum yang dapat dilewatkan pada modul ini sebesar 30 *ampere*. Gambar modul rangkaian driver L298N *Dual H-Bridge* dapat dilihat pada Gambar 4.7.



Gambar 4.7 Modul Rangkaian Driver L298N DUAL H-BRIDGE

Sumber: <http://jualarduinomurah.com/l298n-module>

### 4.3.3 Motor DC

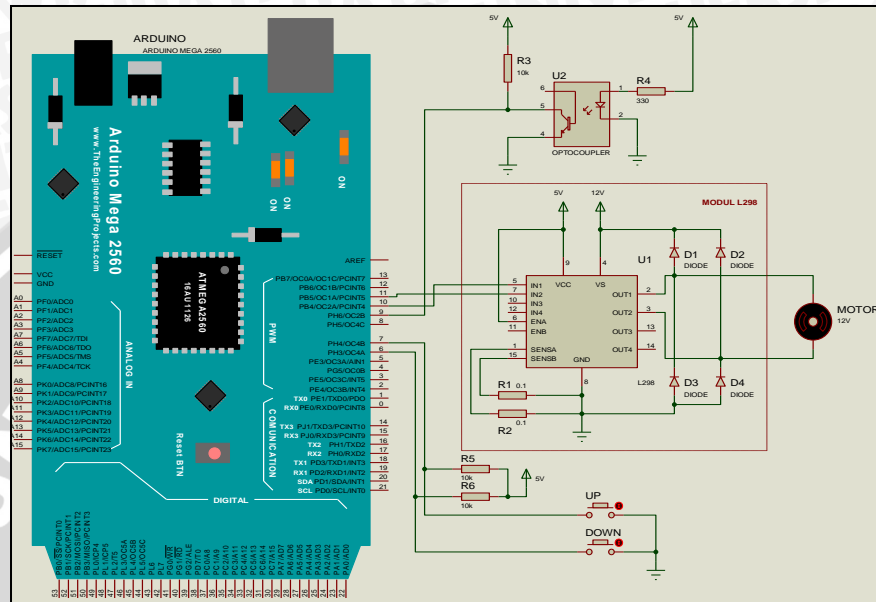
Pengendalian motor DC menggunakan rangkaian Arduino Mega 2560 sebagai pengolah dan memberikan data berupa *Pulse width Modulation*(PWM) agar motor bergerak. Motor DC yang digunakan pada perancangan ini tidak diketahui karakteristiknya, sehingga yang perlu dilakukan adalah melakukan pengujian dengan menggunakan sensor *rotary encoder*. Karakteristik motor DC pada perancangan ini didapatkan dengan cara memberi masukan *unit step*. Hasil kecepatan motor terhadap PWM ditunjukkan dalam Gambar 4.8.



Gambar 4.8 Grafik Karakteristik Motor DC

### 4.3.4 Modul Arduino Mega 2560

Pada model miniatur *towercrane* ini digunakan Arduino Mega 2560 sebagaipengolah dalam proses pengaturan posisi motor DC dalam menggerakkan model miniatur *tower crane*. Konfigurasi kaki I/O dari Arduino Mega 2560 ditunjukkan dalam Gambar 4.9.



Gambar 4.9 Desain Sistem Arduino Mega 2560

Arduino Mega 2560 adalah merupakan *board* mikrokontroler berbasis ATmega2560. Modul ini memiliki 54 digital *input/output* di mana 14 digunakan untuk *output* PWM dan 16 digunakan sebagai analog *input*, 4 untuk UART, 16 MHz osilator kristal, koneksi USB, *power jack*, ICSP Header, dan tombol reset.

Modul ini memiliki segalanya yang dibutuhkan untuk memprogram mikrokontroler seperti kabel USB dan sumber daya melalui Adaptor ataupun baterai. Pin masukan dan keluaran Arduino Mega 2560 pada perancangan ini akan difungsikan sesuai Tabel 4.1

Tabel 4.1 Fungsi Pin Arduino Mega 2560

No	Pin	Fungsi
1	2	Masukan <i>rotary encoder A</i>
2	3	Masukan <i>rotary encoder B</i>
3	10	Masukan <i>enable 1 motor</i>
4	11	Masukan <i>enable 2 motor</i>

5	6	Masukan <i>Push button UP</i>
6	7	Masukan <i>Push button DOWN</i>
7	9	Masukan <i>Optocoupler</i>
8	GND	Jalur masukan GND seluruh sistem
19	Vin	Jalur masukan 5V seluruh sistem

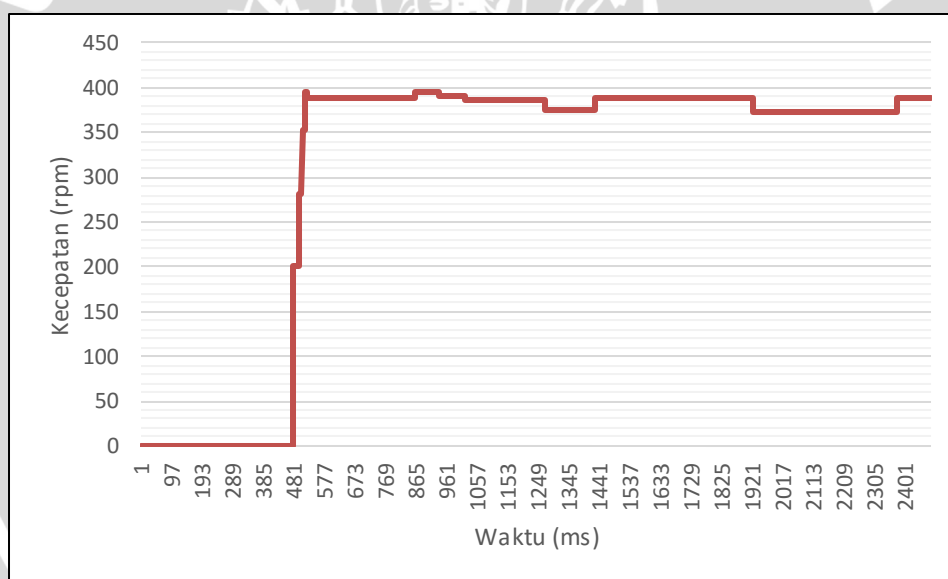
**Perancangan**

**4.3.5**

**Kontroler**

Untuk memenuhi tujuan performansi *loop* yang diinginkan, maka perlu ditambahkan kontroler pada sistem tersebut. Kontroler yang dipilih adalah kontroler Proporsional Integral Diferensial (PID). Kontroler proporsional integral diferensial dipilih karena karakteristik *plant* tersebut membutuhkan kontrol yang cepat dan meminimalkan *overshoot* untuk dapat menghasilkan kecepatan *Tower Crane* yang kondusif.

Dalam skripsi ini diketahui karakteristik *plant* tanpa menggunakan kontroler yaitu dengan memberi masukan *unit step* hingga mencapai *setpoint* sebesar 400 RPM.



**Gambar 4.10 Grafik Respon Sistem Tanpa Kontroler**

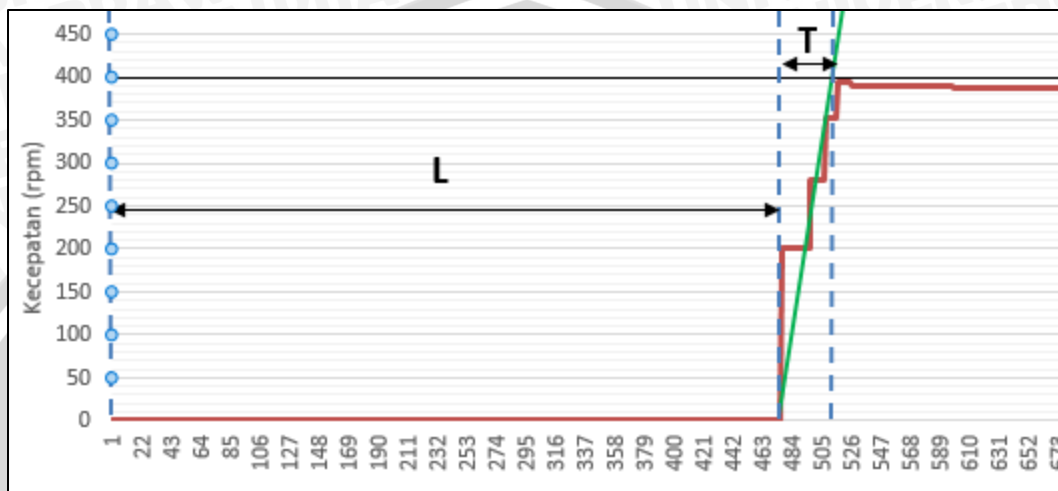
Waktu yang diperlukan alat tersebut untuk mencapai *set point* 400 RPM adalah 516.2 ms. Grafik *plant* dalam Gambar 4.10 menunjukkan bahwa bentuk kurva karakteristik *plant* menyerupai huruf S (*S-shaped curve*) sehingga dapat digunakan *tuning* Kp, Ki, dan Kd menggunakan metode 1 Ziegler-Nichols.

Dari respon yang didapat, sistem memiliki *error steady state* yang besar sehingga diperlukan kontroler untuk memperbaiki sistem. Setelah mendapatkan grafik respon dari

sistem tanpa kontroler, maka langkah selanjutnya adalah menentukan parameter kontroler PID berdasarkan metode 1 Ziegler-Nichols (metode loop tertutup).

#### 4.3.6 Perancangan Kontroler PID metode pertama Ziegler-Nichols

Setelah mendapatkan bentuk kurva S secara eksperimental kemudian dilakukan langkah untuk menentukan nilai L dan T seperti yang pada Gambar 4.11 berikut:



Gambar 4.11 Metode 1 Ziegler-Nichols (Perancangan)

Langkah menentukan *tuning* parameter dengan metode 1 Ziegler-Nichols adalah sebagai berikut:

1. Menarik garis *tangent* pada titik infleksi grafik karakteristik *plant* seperti pada Gambar 4.11.
2. Menentukan perpotongan garis *tangent* terhadap sumbu waktu *t* untuk mendapatkan nilai L.
3. Menentukan perpotongan garis *tangent* terhadap sumbu *steady* untuk mendapatkan nilai T.
4. Setelah mendapatkan nilai L dan T kemudian menentukan nilai  $K_p$ ,  $T_i$  dan  $T_d$  sesuai ketentuan pada tabel 4.2.

Tipe Kontrol	$K_p$	$T_i$	$T_d$
P	$\frac{T}{L}$	$\infty$	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	$2L$	$0,5 L$

**Tabel 4.2 Aturan Metode 1 Ziegler-Nichols (Ogata K., 1997)**

Berdasarkan Gambar 4.11 diperoleh nilai  $L = 480$  dan nilai  $T = 36,2$  sehingga dapat diketahui :

$$Kp = 1,2 \frac{T}{L} = 1,2 \frac{36,2}{480} = 0,091$$

$$Ti = 2L = 2 \times 480 = 960$$

$$Td = 0,5L = 0,5 \times 480 = 240$$

Setelah mendapatkan nilai  $Ti$  dan  $Td$  maka selanjutnya mencari nilai  $Ki$  dan  $Kd$  dengan cara :

$$Ki = \frac{Kp}{Ti} = \frac{0,091}{960} = 0,003834$$

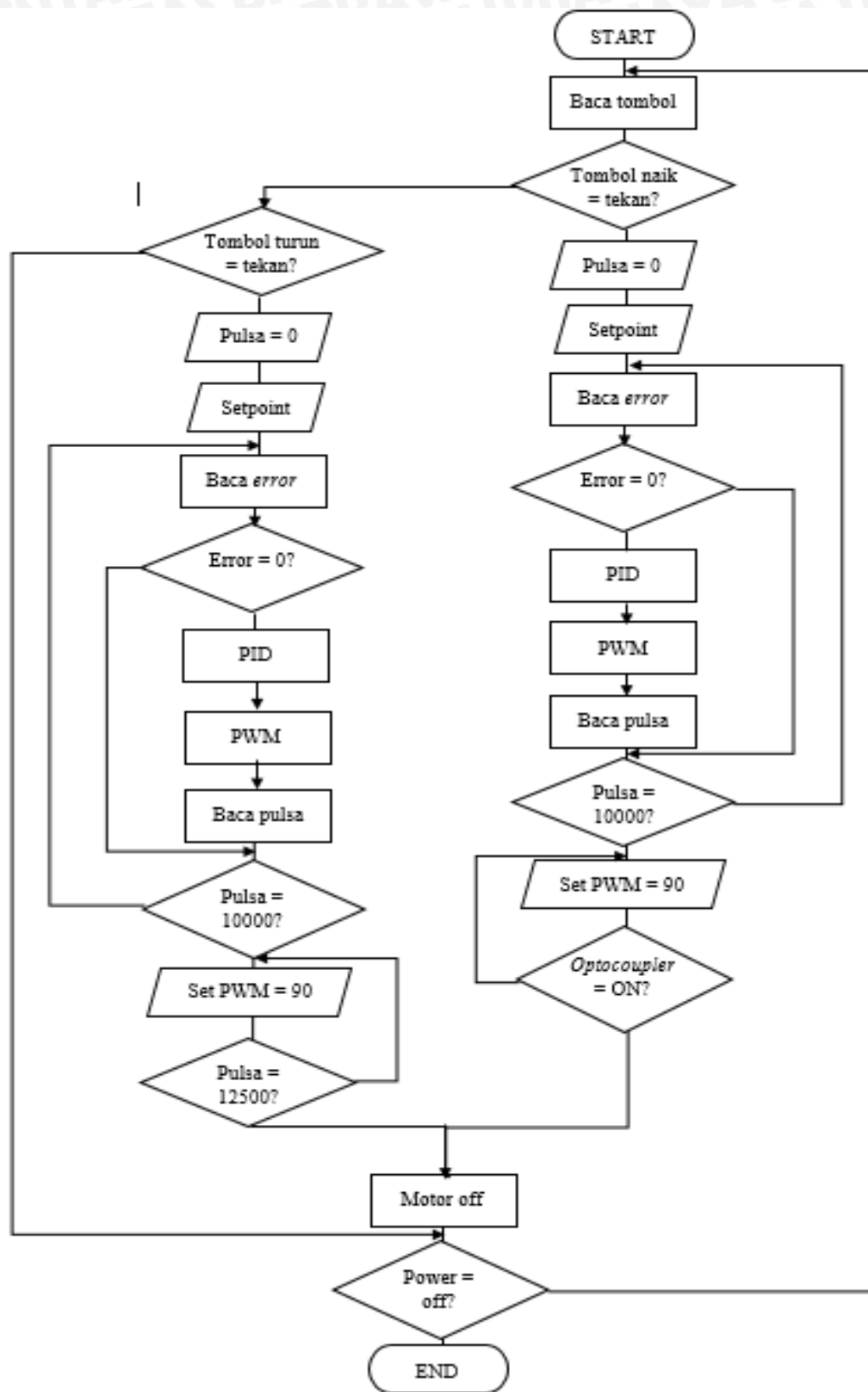
$$Kd = Kp \times Td = 0,091 \times 240 = 0,02172$$

Sehingga telah didapatkan  $Kp$ ,  $Ki$  dan  $Kd$  melalui metode 1 Ziegler-Nichols sebesar  $Kp = 0,091$ ,  $Ki = 0,003834$ , dan  $Kd = 0,02172$

#### **4.4 Perancangan Perangkat Lunak**

##### **4.4.1 Flowchart Sistem Keseluruhan**

Flowchart keseluruhan sistem ditunjukkan pada Gambar 4.12.



Gambar 4.12 Flowchart Keseluruhan Sistem





## BAB V

### PENGUJIAN DAN ANALISIS SISTEM

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

1. Pengujian sensor *rotary encoder*
2. Pengujian *driver* motor DC
3. Pengujian kecepatan motor DC
4. Pengujian tanpa kontroler
5. Pengujian keseluruhan sistem

#### a. Pengujian Sensor *Rotary Encoder*

Pengujian sensor kecepatan dilakukan untuk mengetahui RPM yang terbaca pada sensor kecepatan *rotary encoder autonic* E40H8 500-6-L-5 telah sesuai dengan yang diinginkan.

##### • Peralatan yang digunakan

1. Catu daya 5 V dan 12 V
2. Sensor kecepatan *rotary encoder autonic* E40H8 500-6-L-5
3. Motor *Direct Current* (DC)
4. Program dan *software* Arduino
5. *Driver* motor
6. *Tachometer*

##### • Prosedur Pengujian

1. Merangkai peralatan yang digunakan.
2. Program kecepatan berupa *Pulse Width Modulation* (PWM) dimasukkan ke dalam Arduino.
3. Mengaktifkan catu daya 5 V dan 12 V.

4. Mencatat data kecepatan yang diperoleh dari pembacaan serial monitor dan *tachometer*.
5. Membandingkan data kecepatan dari kedua pembacaan.

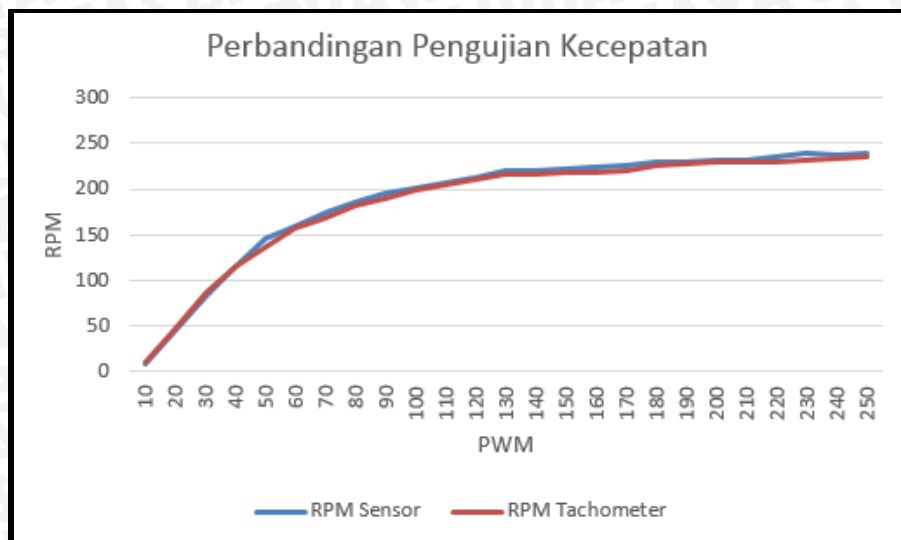
- **Hasil Pengujian**

Setelah melakukan prosedur pengujian, didapatkan data hasil pengujian motor *Direct Current* (DC) dengan nilai *Pulse Width Modulation* (PWM) yang ditentukan. Data hasil pengujian ditunjukkan pada Tabel 5.1 berikut.

**Tabel 5.1 Hasil Perbandingan Nilai Pembacaan Sensor Dan *Tachometer***

PWM	RPM Sensor	RPM <i>Tachometer</i>
10	9,24	10
20	45,12	46
30	82,68	86
40	114,72	116
50	144,96	136
60	159,36	158
70	173,88	168
80	185,4	182
90	194,52	190
100	201,6	200
110	207,24	204
120	213	210
130	220,08	216
140	220,36	216
150	222	218
160	224,52	218
170	226,68	220
180	229,32	226
190	229,56	228
200	231	230
210	232,32	230
220	235,2	230
230	238,8	232
240	237,48	234
250	239,64	236
255	243,72	240

Dari tabel 5.1 diperoleh grafik hubungan antara kecepatan dengan *Pulse Width Modulation* (PWM) seperti pada Gambar 5.1.



Gambar 5.1 Grafik Perbandingan Nilai Pembacaan Sensor Dengan Tachometer

Dari hasil pengujian yang dilakukan dapat disimpulkan bahwa hasil pengujian sensor *rotary encoder* sesuai dengan data pengukuran secara praktek menggunakan *tachometer* dan sensor mampu berjalan dengan baik.

#### b. Pengujian Driver Motor

Pengujian *driver* motor ini bertujuan untuk mengetahui *output driver* motor yang dibandingkan dengan masukannya yang kemudian dapat diketahui juga hubungan keluaran *Pulse Width Modulation* (PWM) dengan tegangan yang dibutuhkan untuk motor *Direct Current* (DC).

- **Peralatan yang Digunakan**

1. Catu Daya 5V dan 12V
2. *Driver* Motor L298N *Dual H-Bridge*
3. Motor *Direct Current* (DC)
4. Sensor kecepatan *rotary encoder autonic* E40H8 500-6-L-5
5. *Avometer*
6. Program dan *software* Arduino.

- **Prosedur Pengujian**

1. Merangkai peralatan yang digunakan.
2. Program kecepatan berupa *Pulse Width Modulation* (PWM) dimasukkan ke dalam Arduino.
3. Mengaktifkan catu daya 5V dan 12V.

4. Mencatat data kecepatan yang diperoleh dari pembacaan *avometer* pada mikrokontroler dan *driver* motor.
5. Membandingkan data kecepatan dari kedua pembacaan.

- **Hasil Pengujian**

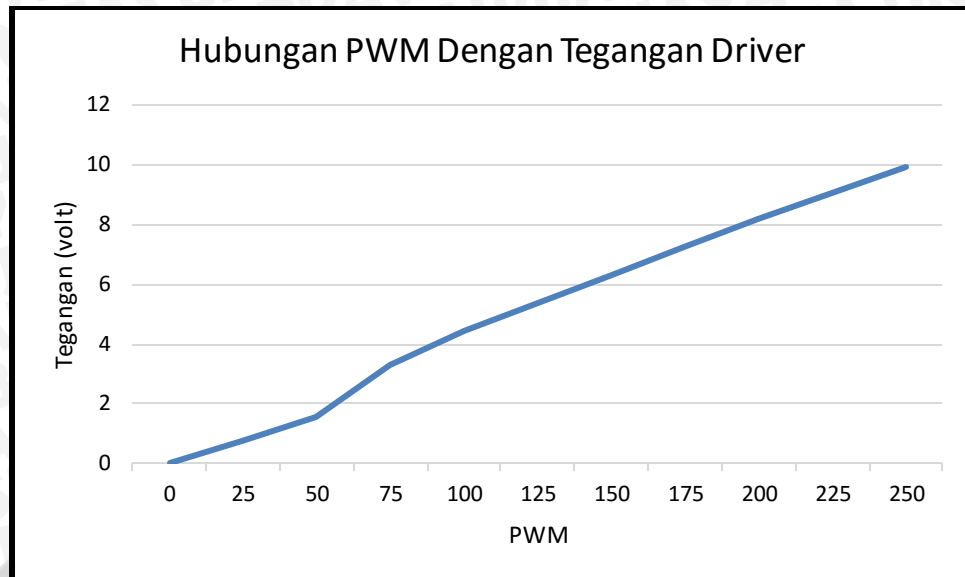
Setelah melakukan prosedur pengujian, didapatkan hasil sinyal dan tampilan parameter. Parameter yang dihasilkan adalah tegangan mikrokontroler dan tegangan *driver*.

Setelah mendapatkan data keluaran dari hasil pengujian *driver* maka dapat diketahui besarnya tegangan yang dibutuhkan motor untuk setiap interval *Pulse Width Modulation* (PWM) yang diberikan. Data yang dihasilkan ditunjukkan pada Tabel 5.2

**Tabel 5.2 Hasil Keluaran Tegangan *Driver* L298N *Dual H-Bridge***

PWM	Tegangan <i>Driver</i>	RPM Sensor
0	0	0
25	0.74	0
50	1.56	0
75	3.32	243.15
100	4.42	410.62
125	5.38	542.36
150	6.33	683.31
175	7.25	795.85
200	8.15	902.15
225	9.02	1001.92
250	9.92	1093.46

Dari tabel 5.2 diperoleh grafik hubungan antara *Pulse Width Modulation* (PWM) dengan keluaran tegangan seperti pada Gambar 5.2.



Gambar 5.2 Grafik Hubungan PWM Dengan Tegangan Keluaran Driver

### c. Pengujian Kecepatan Motor DC

Pengujian kecepatan motor dilakukan untuk mengetahui hubungan PWM dengan kecepatan motor.

- **Peralatan yang digunakan**

1. Catu daya 5 V dan 12 V
2. Sensor kecepatan *rotary encoder autonic* E40H8 500-6-L-5
3. Motor *Direct Current* (DC)
4. Program dan *software* Arduino
5. *Driver* motor L298N *Dual H-Bridge*

- **Prosedur Pengujian**

2. Merangkai peralatan yang digunakan.
3. Mengunduh program kecepatan berupa *Pulse Width Modulation* (PWM) dimasukkan ke dalam Arduino.
4. Mengaktifkan catu daya 5 V dan 12 V.
5. Mencatat data kecepatan yang diperoleh dari pembacaan serial monitor.

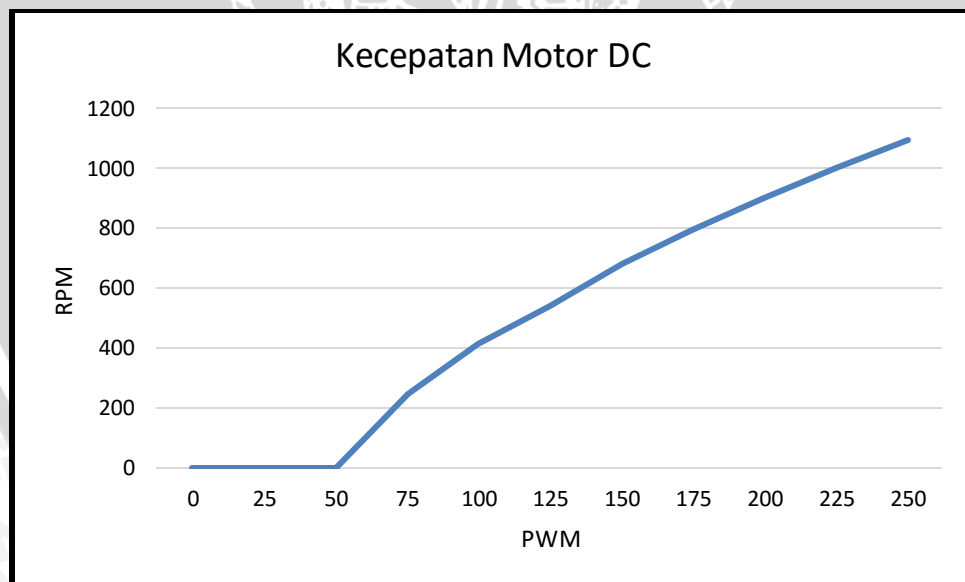
- **Hasil Pengujian**

Setelah melakukan prosedur pengujian, didapatkan data hasil pengujian motor *Direct Current* (DC) dengan nilai *Pulse Width Modulation* (PWM) sebesar 0-255, kemudian diperoleh respon kecepatan motor yang ditunjukkan pada Tabel 5.3 berikut.

**Tabel 5.3 Hasil Keluaran Kecepatan Motor DC**

PWM	RPM Sensor
0	0
25	0
50	0
75	243.15
100	410.62
125	542.36
150	683.31
175	795.85
200	902.15
225	1001.92
250	1093.46

Dari tabel 5.3 diperoleh grafik kecepatan motor DC dengan masukan *Pulse Width Modulation* (PWM) seperti pada Gambar 5.3



**Gambar 5.3 Grafik Hubungan PWM Dengan Kecepatan Motor**

#### d. Pengujian Tanpa Kontroler

Pengujian ini bertujuan untuk mengetahui bagaimana perbedaan respon sistem terhadap *setpoint* ketika tanpa kontroler dengan menggunakan kontroler, sehingga dapat ditentukan bahwa sistem memerlukan sebuah sistem pengontrolan.

- **Peralatan yang digunakan**

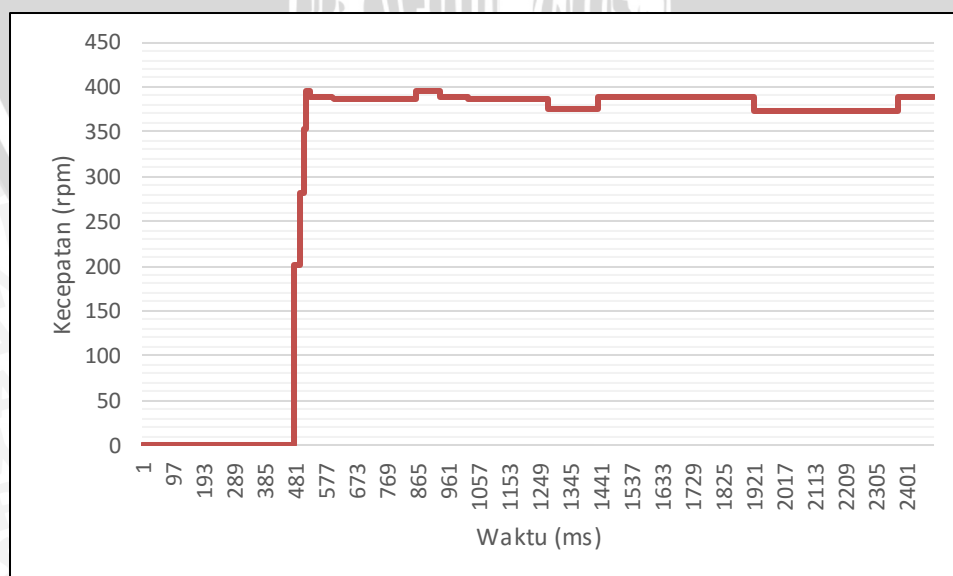
1. Catu daya 5V dan 12V
2. *Software* Matlab
3. Arduino Mega 2560
4. Motor DC
5. Sensor kecepatan *rotary encoder autonic* E40H8 500-6-L-5
6. Komputer
7. Program dan software Arduino.

- **Prosedur Pengujian**

1. Merangkai peralatan yang digunakan.
2. Menyusun program pada *software* Arduino ERW 1.0.6 untuk menampilkan hasil respon system tanpa kontroler.
3. Memberi masukan nilai PWM sebesar 95 PWM agar mencapai *setpoint* sebesar 400 RPM sesuai dengan pengujian *driver* motor.
4. Mengamati hasil keluaran sistem pada *serial monitor software* Arduino ERW 1.0.6.

- **Hasil Pengujian**

Setelah melakukan prosedur pengujian, didapatkan hasil respon sistem tanpa kontroler. Grafik hasil pengujian ditunjukkan pada Gambar 5.4.



Gambar 5.4 Pengujian Sistem tanpa kontroler dengan Masukan *Unit Step*



Dalam grafik pada Gambar 5.4, sistem masih membutuhkan waktu yang lama untuk mencapai nilai dari *setpoint* yang ditentukan. dan sistem tidak stabil setelah mencapai *setpoint* maka diperlukan sebuah kontroler untuk menjaga kestabilan sistem agar kecepatan tetap pada posisi *setpoint*.

### 5.5. Pengujian Sistem Secara Keseluruhan

Pengujian dilakukan untuk mengetahui bagaimana respon keseluruhan sistem pengendalian kecepatan motor DC pada model miniatur *tower crane* dapat bekerja dengan baik sesuai dengan *setpoint* yang diharapkan.

- **Peralatan yang digunakan**

1. Model miniatur *tower crane*.
2. Catu daya 5 V dan 12 V
3. Sensor kecepatan *rotary encoder autonic* E40H8 500-6-L-5
4. Sensor *Optocoupler*.
5. Motor *Direct Current* (DC)
6. Program dan *software* Arduino
7. *Driver* motor L298N *Dual H-Bridge*.

- **Prosedur Pengujian**

Pengujian alat ini dilakukan dengan cara menjalankan model miniatur *tower crane* dengan cara memasukkan nilai *set point* kecepatan motor sebesar 400 RPM dengan parameter PID yang dicari menggunakan metode 1 *Ziegler-nichols*  $K_p = 0,091$ ,  $K_i = 0,003834$ ,  $K_d = 0,02172$  pada *software* arduino. Setelah itu mengamati hasil keluaran berupa nilai RPM yang terbaca kemudian membuat grafik kecepatan terhadap waktu yang menampilkan data setiap 1ms. Pengujian pada sistem dilakukan tanpa menggunakan beban dan dengan menggunakan beban maksimal seberat 200 gram. Langkah terakhir mencatat *error* yang menjadi hasil pembacaan sensor dengan *setpoint* yang diinginkan. Penyusunannya seperti yang ditunjukkan gambar 5.5.



Gambar 5.5 Model Miniatur *Tower Crane* secara keseluruhan

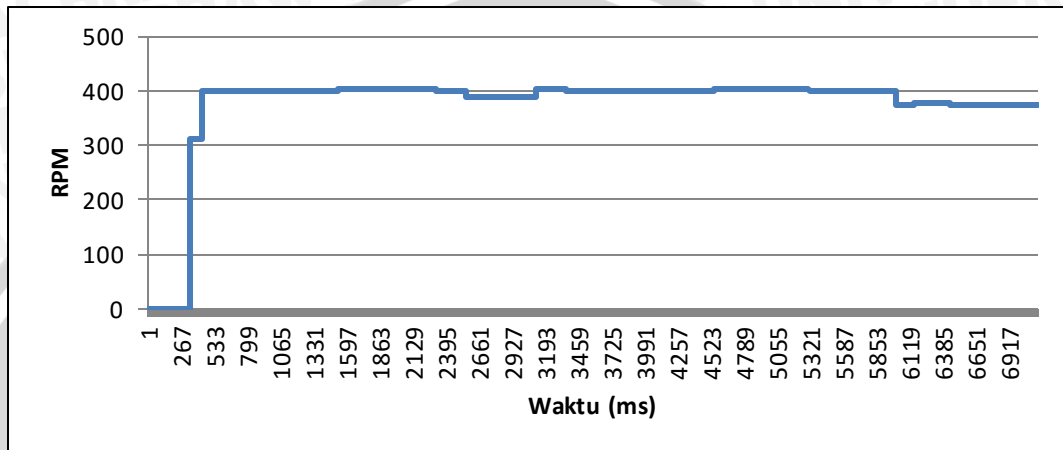


- **Hasil Pengujian**

Setelah dilakukan pengujian, didapatkan hasil respon keluaran sistem sebagai berikut:

- a. Pengujian *tower crane* bergerak naik tanpa beban

Grafik Respon Sitem Keseluruhan saat *tower crane* bergerak naik tanpa beban ditunjukkan pada Gambar 5.6.



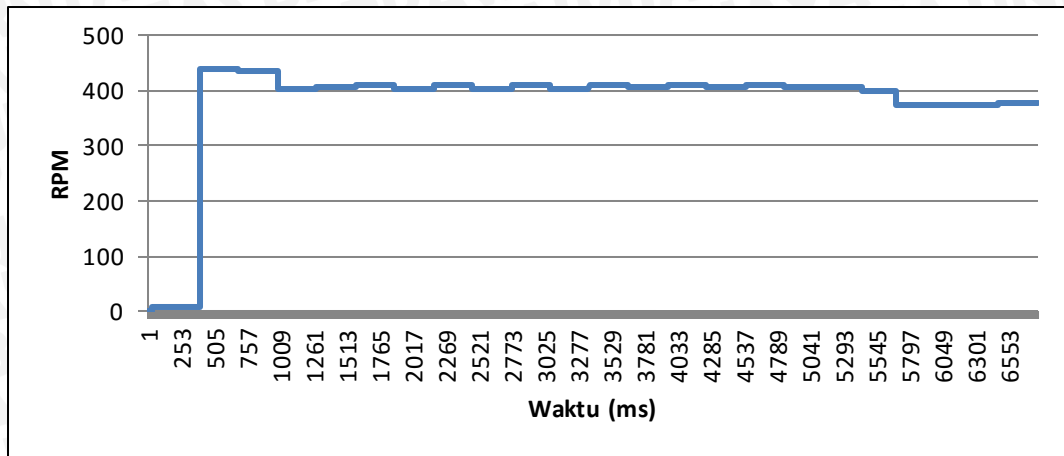
Gambar 5.6 Grafik Respon Sitem Keseluruhan (saat *tower crane* naik tanpa beban)

Dari grafik Gambar 5.6, diketahui bahwa hasil respon memiliki waktu *settling time* 448 ms untuk mencapai *steady state* dan mengalami proses pengereman pada waktu 6025 ms. Serta memiliki % *error* sebagai berikut:

$$\begin{aligned}
 \% \text{Ess} &= \frac{1}{N} \sum_{i=1}^N \left| \frac{\text{output } [i] - \text{setpoint}}{\text{setpoint}} \right| \times 100\% \\
 &= \frac{1}{448} \sum_{i=1}^N 0,0008 \times 100\% \\
 &= 0,22 \%
 \end{aligned}$$

- b. Pengujian *tower crane* bergerak turun tanpa beban

Grafik Respon Sitem Keseluruhan saat *tower crane* bergerak turun tanpa beban ditunjukkan pada Gambar 5.7.



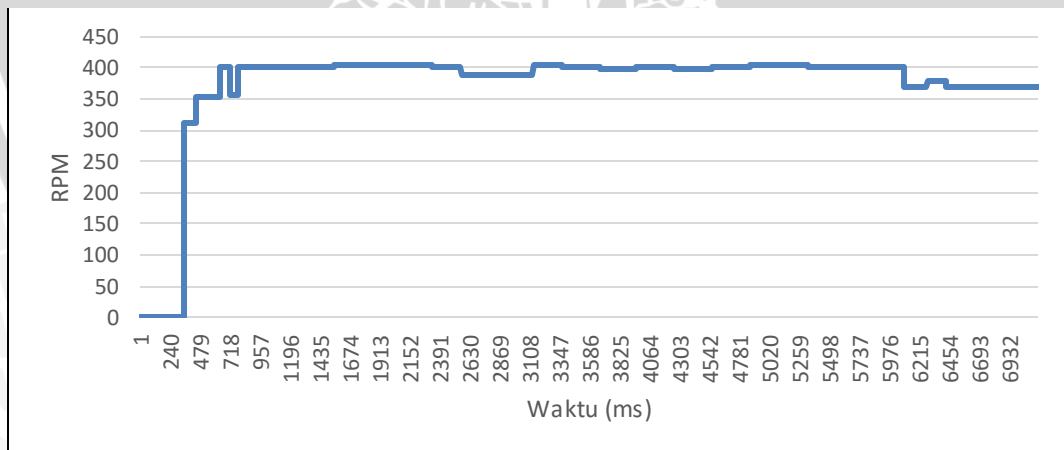
Gambar 5.7 Grafik Respon Sitem Keseluruhan (saat tower crane turun tanpa beban)

Dari grafik Gambar 5.7, diketahui bahwa hasil respon memiliki waktu *settling time* 997 ms untuk mencapai *steady state* dan mengalami proses pengereman pada waktu 5705 ms. Serta memiliki % *error* sebagai berikut:

$$\begin{aligned}
 \% \text{Ess} &= \frac{1}{N} \sum_{i=1}^N \left| \frac{\text{output } [i] - \text{setpoint}}{\text{setpoint}} \right| \times 100\% \\
 &= \frac{1}{997} \sum_{i=1}^N 0,017 \times 100\% \\
 &= 0,1 \%
 \end{aligned}$$

c. Pengujian *tower crane* bergerak naik dengan beban maksimal

Grafik Respon Sitem Keseluruhan saat *tower crane* bergerak naik dengan beban maksimal ditunjukkan pada Gambar 5.8.



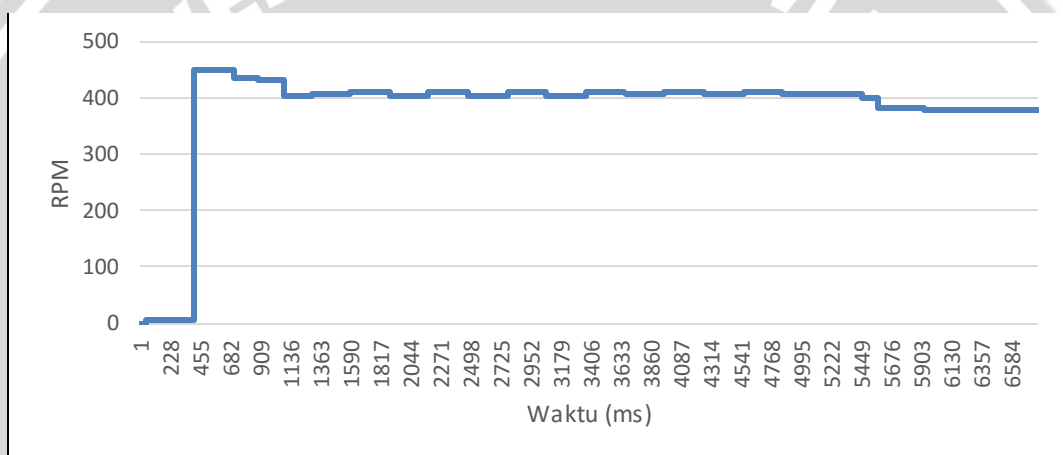
Gambar 5.8 Grafik Respon Sitem Keseluruhan (saat tower crane naik dengan beban maksimal)

Dari grafik Gambar 5.8, diketahui bahwa hasil respon memiliki waktu *settling time* 639 ms untuk mencapai *steady state* dan mengalami proses pengereman pada waktu 6101 ms. Serta memiliki % *error* sebagai berikut:

$$\begin{aligned} \% \text{Ess} &= \frac{1}{N} \sum_{i=1}^N \left| \frac{\text{output } [i] - \text{setpoint}}{\text{setpoint}} \right| \times 100\% \\ &= \frac{1}{639} \sum_{i=1}^N \times 0,02\% \\ &= 0,15 \% \end{aligned}$$

d. Pengujian *tower crane* bergerak turun dengan beban maksimal

Grafik Respon Sitem Keseluruhan saat *tower crane* bergerak turun dengan beban maksimal ditunjukkan pada Gambar 5.9.



Gambar 5.9 Grafik Respon Sitem Keseluruhan (saat *tower crane* turun dengan beban maksimal)

Dari grafik Gambar 5.9, diketahui bahwa hasil respon memiliki waktu *settling time* 1079 ms untuk mencapai *steady state* dan mengalami proses pengereman pada waktu 5583 ms. Serta memiliki % *error* sebagai berikut:

$$\begin{aligned} \% \text{Ess} &= \frac{1}{N} \sum_{i=1}^N \left| \frac{\text{output } [i] - \text{setpoint}}{\text{setpoint}} \right| \times 100\% \\ &= \frac{1}{1079} \sum_{i=1}^N 0,018 \times 100\% \\ &= 0.094 \% \end{aligned}$$

Dari pengujian keseluruhan dapat diketahui bahwa *settling time* terbesar terjadi saat *tower crane* bergerak turun dengan beban maksimal. Hal ini disebabkan oleh kekuatan mekanik karena motor DC terjadi *loss*.



## BAB VI

### I. PENUTUP

#### 6.1 Kesimpulan

Dari perancangan, pengujian dan pengamatan yang telah dilakukan pada penelitian sistem pengendalian kecepatan maka dapat diambil kesimpulan sebagai berikut :

1. Berdasarkan data respon sistem yang diperoleh dari pengujian dengan menggunakan metode pertama *Ziegler-Nichols*, maka parameter kontroler PID dapat ditentukan dengan gain  $K_p = 0,091$ ,  $K_i = 0.003834$  dan  $K_d = 0.02172$ .
2. Hasil pengujian terhadap sistem pengendalian kecepatan putaran motor DC pada *tower crane* menunjukkan bahwa respon sistem memiliki *error stady state* kurang dari 5 % dan mengalami *settling time* kurang dari 1000 ms. Tetapi pada keadaan turun dengan beban maksimal, terjadi kerusakan mekanik sehingga mengalami *settling time* yang cukup besar.

#### 6.2 Saran

Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan. Untuk memperbaiki kinerja alat dan pengembangan lebih lanjut disarankan:

1. Disarankan untuk menggunakan motor DC yang *high torque* dan pengkopelan *gear* dengan baik agar tidak terjadi *loss* pada motor.
2. Disarankan untuk memasang sensor *rotary encoder* pada as motor DC secara langsung, tanpa menggunakan *gear*.
3. Disarankan untuk melakukan pengontrolan dengan metode selain PID.

UNIVERSITAS BRAWIJAYA





## DAFTAR PUSTAKA

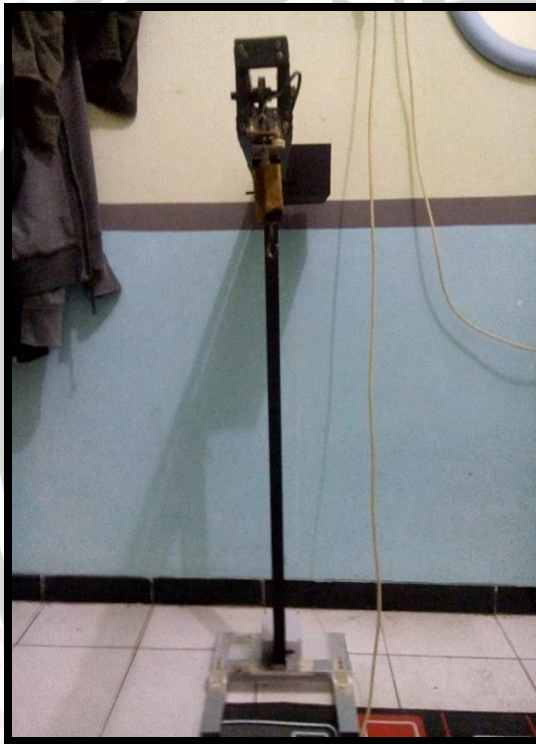
- Barley, Eres. 2011. *Sistem Pengaturan Kecepatan Putaran Motor DC Pada Konveyor Bertingkat Menggunakan Kontroler PID Berbasis PLC*. Malang: Universitas Brawijaya.
- Basori, Slamet. 2014. *Implementasi Odometry pada Robot Otomatis Kontes Robot ABU Indonesia*. Malang: Universitas Brawijaya.
- Braunl, Thomas. 2003. *Embedded Robotics*. Jerman: Springer.
- Juwana, Jimmy. 2005. *Panduan Sistem Bangunan Tinggi*. Jakarta. Penerbit Erlangga
- Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik Jilid 1*. Jakarta. Penerbit Erlangga.
- Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik Jilid 2*. Jakarta. Penerbit Erlangga.
- Ogata, K. 2004. *System Dynamics 4th edition*. New York : Prentice-Hall, Inc.
- Ogata, K. 2002. *Modern Control Engineering*. New Jersey : Prentice-Hall, Inc.
- Rifky Nugraha, Arga. 2014. *Sistem Pengaturan Kecepatan Motor DC pada Alat Pengaduk Adonan Dodol Menggunakan Kontroler PID*. Malang: Universitas Brawijaya.
- Soemarwanto. 1999. *Dasar Konversi Energi Elektrik Jilid II*. Malang: Fakultas Teknik Universitas Brawijaya.

UNIVERSITAS BRAWIJAYA



# LAMPIRAN

## Foto Alat



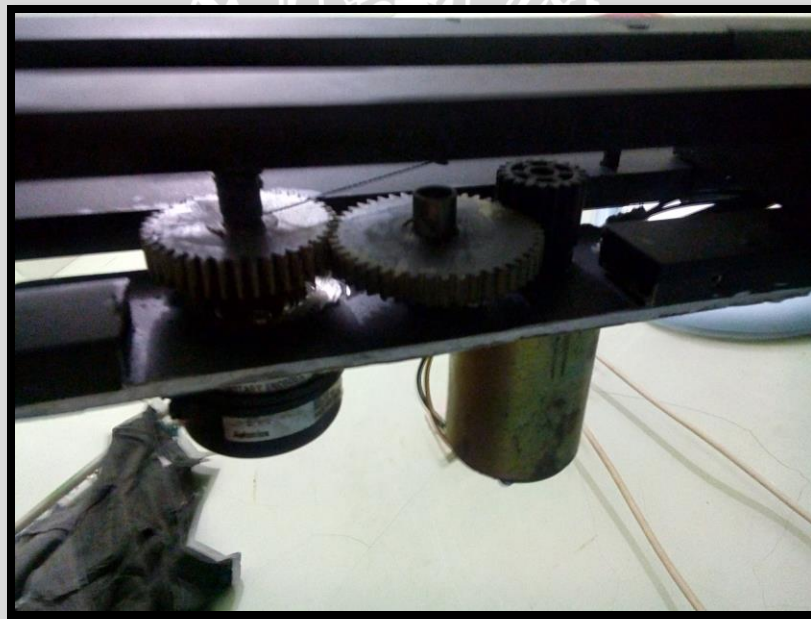
Alat tampak depan



Alat tampak samping



Rangkaian elektrik pada alat



Motor DC dan *Rotary encoder*



Penempatan sensor *optocoupler*



Alat tampak keseluruhan

# Listing Program

```
#include <PID_v1.h>
#include <TimerOne.h>
#include <TimerThree.h>

#define PIN_INPUT 0
#define PIN_OUTPUT 3
#define EN1 10
#define EN2 12
#define pin_PWM 11

#define ENCODER_B 3
#define TOP_LIMITER 9
#define TOMBOL_UP 6
#define TOMBOL_DOWN 7

#define TEKAN 0
#define LEPAS 1

// #include <TimerOne.h>

double Setpoint, Input, Output;

double consKp=0.091, consKi= 0.003834, consKd= 0.02172;

bool encoder_state_A=false;
// tuning parameters PID berdasarkan library
PID myPID(&Input, &Output, &Setpoint, consKp, consKi, consKd, DIRECT);
double pulsa=0;
float RPM,clock=0;

byte pwmn=0;
void pulse_A()
{
    encoder_state_A=true;
}

void pulse_B()
{
    if (encoder_state_A==true)
    {
        pulsa++;
        clock++;
        encoder_state_A=false;
    }
}

void setup()
```

```
{
  pinMode(TOP_LIMITER, INPUT);
  pinMode(TOMBOL_UP, INPUT_PULLUP);
  pinMode(TOMBOL_DOWN, INPUT_PULLUP);
  pinMode(EN1, OUTPUT);
  pinMode(EN2, OUTPUT);
  pinMode(pin_PWM,OUTPUT);
  Serial.begin(115200);
  attachInterrupt(0, pulse_A, FALLING); //INT5
  attachInterrupt(1, pulse_B, FALLING); //INT5
  Serial.println("TES");
  //PID on
  myPID.SetMode(AUTOMATIC);
  myPID.SetOutputLimits(0, 255);
  myPID.SetControllerDirection(DIRECT);
  myPID.SetSampleTime(1);

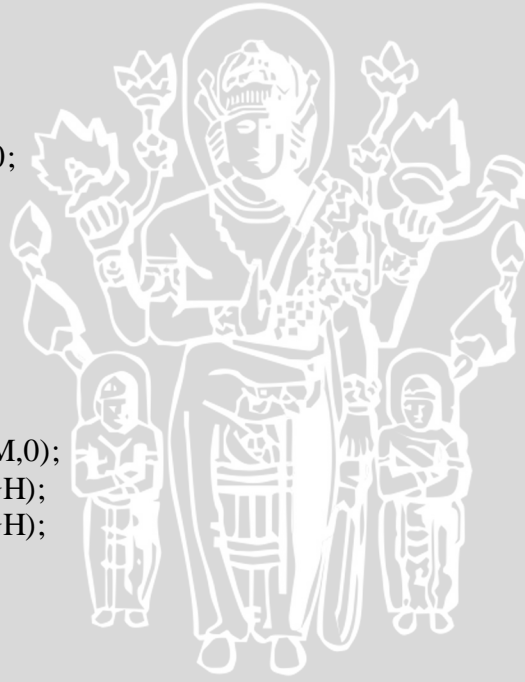
  Timer3.initialize(250000);
  Timer3.attachInterrupt( cek_RPM ); // attach the service routine here
}

void cek_RPM()
{
  float hitung=clock/260;
  RPM=(hitung*4)*60;
  Input=RPM;
  clock=0;
}

void gondola_stop()
{
  analogWrite(pin_PWM,0);
  digitalWrite(EN1,HIGH);
  digitalWrite(EN2,HIGH);
}

void gondola_naik()
{
  analogWrite(pin_PWM,0);
  digitalWrite(EN1,HIGH);
  digitalWrite(EN2,LOW);
}

void gondola_turun()
{
  analogWrite(pin_PWM,0);
  digitalWrite(EN1,LOW);
  digitalWrite(EN2,HIGH);
}
```

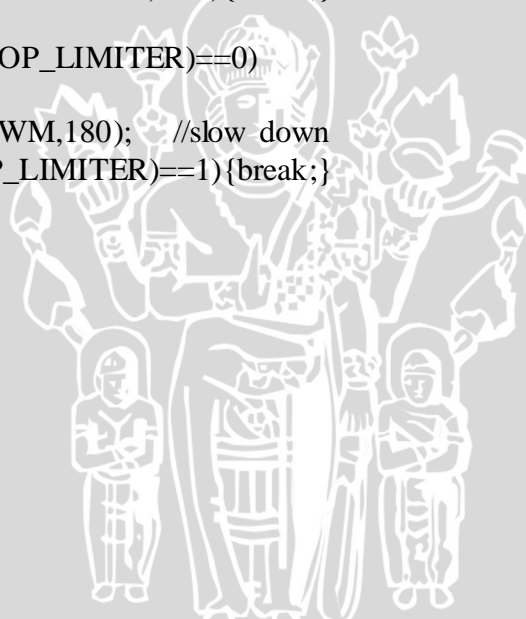


```
void tarik_gondola()
{
  Setpoint=400;
  pulsa=0;
  gondola_naik();
  while(pulsa<10000)
  {
    Input=RPM;
    myPID.SetTunings(consKp, consKp, consKd); //ms
    myPID.Compute();
    Serial.println(RPM);
    int mypwm = map(Output, 0, 255, 100, 255);

    analogWrite(pin_PWM, mypwm);
    if (digitalRead(TOP_LIMITER)==1){break;}
  }
  while(pulsa<12500)
  {
    analogWrite(pin_PWM, 100); //slow down
    if (digitalRead(TOP_LIMITER)==1){break;}
  }
  while(digitalRead(TOP_LIMITER)==0)
  {
    analogWrite(pin_PWM, 180); //slow down
    if (digitalRead(TOP_LIMITER)==1){break;}
  }
  gondola_stop();
}

void turun_gondola()
{
  Setpoint=400;
  pulsa=0;
  gondola_turun();
  while(pulsa<10000)
  {
    Input=RPM;
    myPID.SetTunings(consKp, consKp, consKd); //ms
    myPID.Compute();
    Serial.println(RPM);

    int mypwm = map(Output, 0, 255, 100, 255);
    analogWrite(pin_PWM, mypwm);
    digitalWrite(EN1, LOW);
    if (digitalRead(TOP_LIMITER)==1){break;}
  }
  while(pulsa<12500)
  {
    analogWrite(pin_PWM, 100); //slow down
  }
}
```



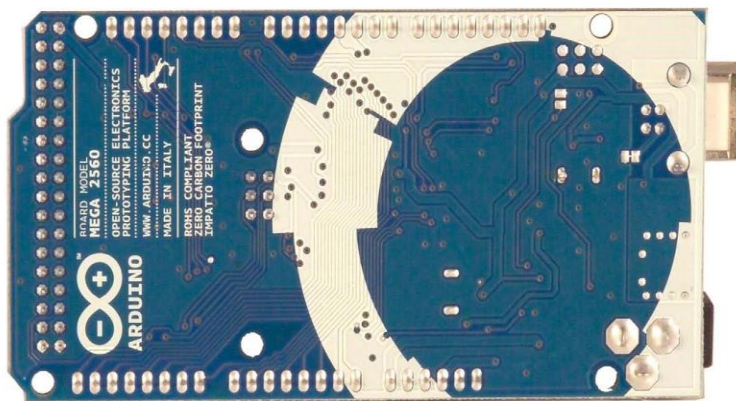
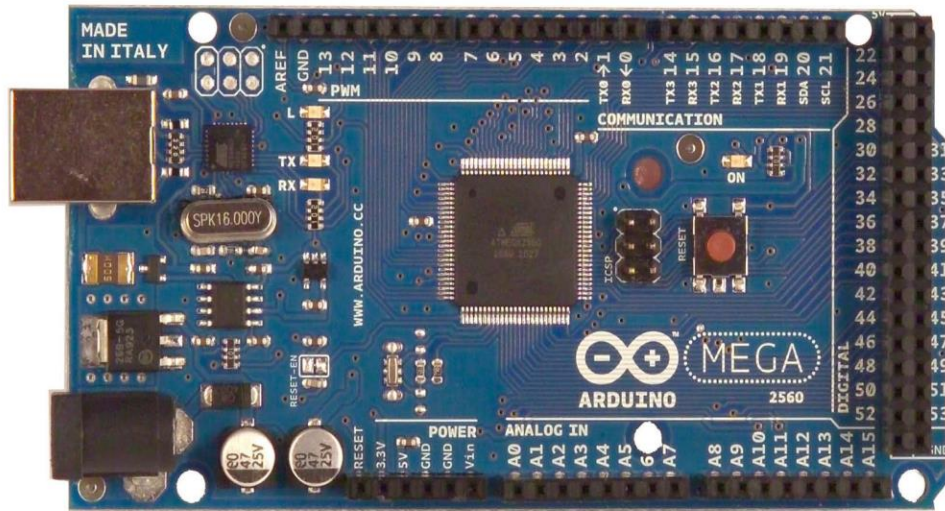


```
gondola_stop();  
}  
  
void loop()  
{  
  if(digitalRead(TOMBOL_UP)==0)  
  {  
    tarik_gondola();  
  }  
  else if(digitalRead(TOMBOL_DOWN)==0)  
  {  
    turun_gondola();  
  }  
}
```



# Datasheet

## Arduino Mega 2560



The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 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 is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

## II. Schematic & Reference Design

EAGLE files: [arduino-mega2560-reference-design.zip](#)

Schematic: [arduino-mega2560-schematic.pdf](#)

## III. Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	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

## IV. Power

The Arduino Mega 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 be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- + **VIN**. The input voltage to the Arduino 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**. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- + **3V3**. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- + **GND**. Ground pins.

## V. 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](#)).

## VI. Input and Output

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 a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. 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 ATmega8U2 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 value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.

- + **PWM: 0 to 13**. Provide 8-bit PWM output with the `analogWrite()` function.

- + **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS)**. These pins support SPI communication using the `SPI library`. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.

- + **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.

+ **I<sup>2</sup>C: 20 (SDA) and 21 (SCL)**. Support I<sup>2</sup>C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I<sup>2</sup>C pins on the Duemilanove or Diecimila.

The Mega2560 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 it is 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.

## VII. Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 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 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 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 Mega2560's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

## VIII. Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference and tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned 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; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available in the [Arduino repository](#). The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. 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.

### IX. Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 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 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 Mega2560 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 Mega2560. 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 Mega2560 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.

### X. USB Overcurrent Protection

The Arduino Mega2560 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.

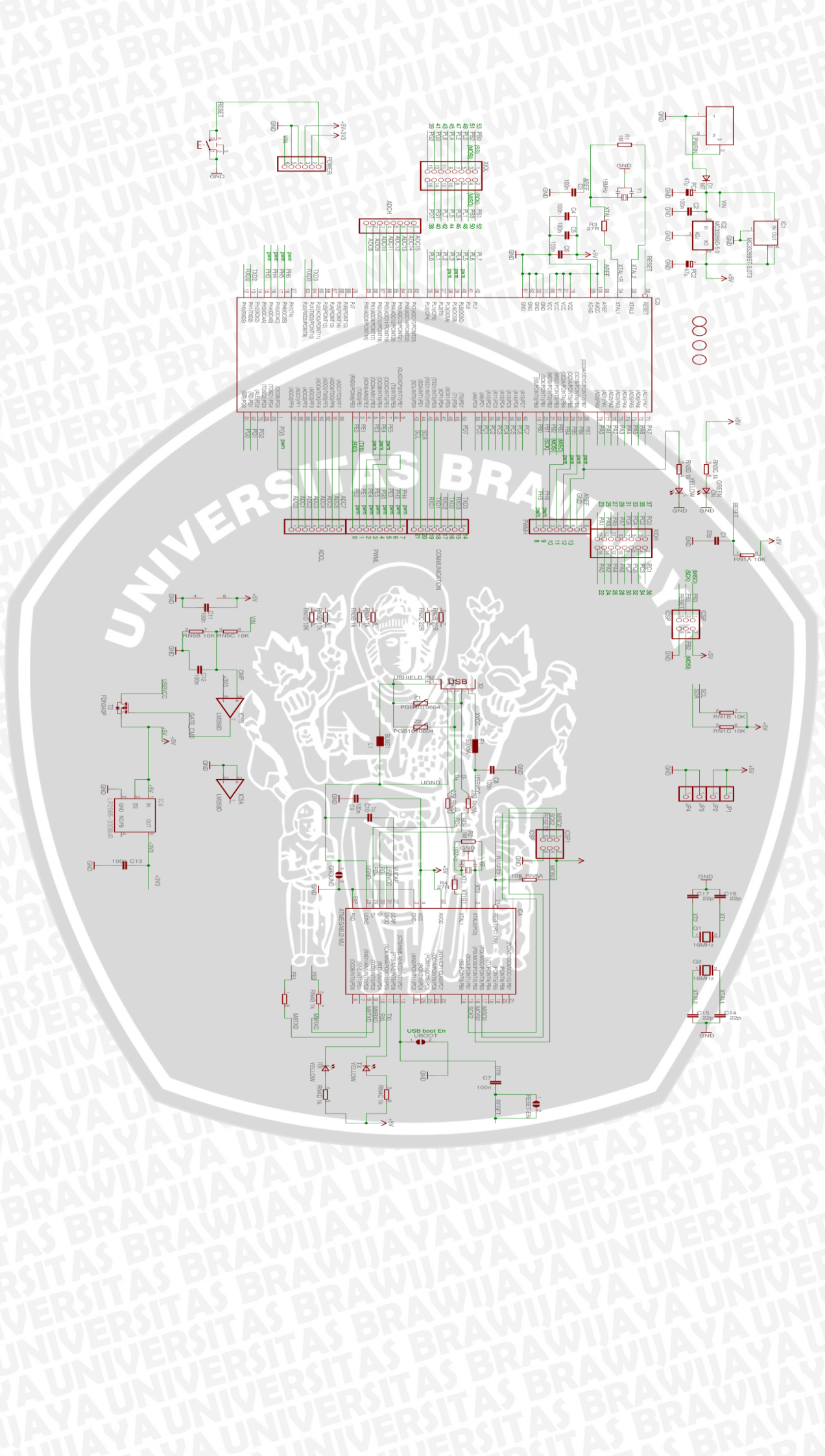
## XI Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega2560 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 Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. 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. Further 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 Mega2560 and Duemilanove/

Diecimila. *Please note that I<sup>2</sup>C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).*







# E40 Series

## Diameter ø40mm Shaft type/Hollow type/Built-in type Incremental Rotary Encoder

### ■ Features

- 12-24VDC power supply of line driver output(Line-up)
- Easy installation at narrow space
- Small moment of inertia
- Power supply : 5VDC, 12-24VDC ±5%
- Various output types

⚠ Please read "Caution for your safety" in operation manual before using.



### ■ Ordering information

E40 [ ] 8 - 5000 - 3 - N - 24 - [ ]

Series	Shaft type	Hollow type	Pulse/1 Revolution	Output phase	Output	Power supply	Cable
B: Shaft type	External	Inner	Series	2: A, B 3: A, B, Z 4: A, $\bar{A}$ , B, $\bar{B}$ 5: A, $\bar{A}$ , B, $\bar{B}$ , Z, $\bar{Z}$	T: Totem pole output N: NPN open collector output V: Voltage output L: Line driver output	5 : 5VDC ±5% 24: 12-24VDC ±5%	No mark: Cable type C: Connector cable type(3x)
H: Hollow type	ø6mm	ø6mm					
HB: Hollow built-in type	ø6mm	ø8mm					
	ø8mm	ø10mm					
		ø12mm					

※Standard: E40B8-[ ]-3-N-24, E40H8-[ ]-3-N-24, E40HB8-[ ]-3-N-24

※Cable length : 250mm

### ■ Specifications

Item	Diameter ø40mm shaft/hollow shaft/hollow built-in type of incremental rotary encoder		
Resolution(P/R) <sup>※1</sup>	-1, -2, -5, 10, -12, 15, 20, 23, 25, 30, 35, 40, 45, 50, 60, 75, 100, 120, 150, 192, 200, 240, 250, 256, 300, 360, 400, 500, 512, 600, 800, 1000, 1024, 1200, 1500, 1800, 2000, 2048, 2500, 3000, 3600, 5000 (Not indicated resolution is customizable.)		
Electrical specification	Output phase	A, B, Z phase(Line driver A, $\bar{A}$ , B, $\bar{B}$ , Z, $\bar{Z}$ phase)	
	Phase difference of output	Phase difference between A and B : $\frac{T}{4} \pm \frac{T}{8}$ (T=1 cycle of A phase)	
	Control output	Totem pole output	* Low - Load current: Max. 30mA, Residual voltage : Max. 0.4VDC * High - Load current: Max. 10mA, Output voltage(Power voltage 5VDC): Min. (Power voltage-2.0)VDC, Output voltage(Power voltage 12-24VDC): Min. (Power voltage-3.0)VDC
		NPN open collector output	Load current : Max. 30mA, Residual voltage : Max. 0.4VDC
		Voltage output	Load current : Max. 10mA, Residual voltage : Max. 0.4VDC
	Response time (Rise/Fall)	Line driver output	* Low - Load current : Max. 20mA, Residual voltage : Max. 0.5VDC * High - Load current : -20mA, Output voltage(Power voltage 5VDC): Min. 2.5VDC, Output voltage(Power voltage 12-24VDC): Min. (Power voltage-3.0)VDC
		Totem pole output	Max. 1µs
		NPN open collector output	
		Voltage output	Max. 0.5µs
		Line driver output	
Max. Response frequency	300kHz		
Power supply	* 5VDC ±5%(Ripple P-P : Max. 5%) * 12-24VDC ±5%(Ripple P-P : Max. 5%)		
Current consumption	Max. 80mA(disconnection of the load), Line driver output : Max. 50mA(disconnection of the load)		
Insulation resistance	Min. 100MΩ(at 500VDC megger between all terminals and case)		
Dielectric strength	750VAC 50/60Hz for 1 minute:(Between all terminals and case)		
Connection	Cable type, 250mm connector cable type		
Mechanical specification	Starting torque	Shaft type : Max. 40gf·cm(0.004N·m), Hole type : Max. 50gf·cm(0.005N·m)	
	Moment of inertia	Max. 40g·cm <sup>2</sup> (4×10 <sup>-6</sup> kg·m <sup>2</sup> )	
	Shaft loading	Radial : Max. 2kgf, Thrust : Max. 1kgf	
	Max. allowable revolution <sup>※2</sup>	5000rpm	
Vibration	1.5mm amplitude or 300m/s <sup>2</sup> at frequency of 10 to 55Hz(for 1 min.) in each of X, Y, Z directions for 2 hours		
Shock	Approx. Max. 50G		
Environment	Ambient temperature	-10 to 70°C, storage : -25 to 85°C	
	Ambient humidity	35 to 85%RH, storage : 35 to 90%RH	
Protection	IP50(IEC standard) ※Option type is available for IP64 (IEC standard).		
Cable	ø5, 5-wire, Length : 2m, Shield cable(Line driver output : ø5, 3-wire) (AWG24, Core diameter: 0.08, Number of cores: 40, Insulator out diameter: ø1)		
Accessory	* Shaft type : ø6mm coupling standard, ø8mm coupling(Bold separately) * Hole type : Bracket		
Approval	CE (Except line driver output)		
Unit weight	Approx. 160g		

※1: ~ pulse is only for A, B phase/Line Driver output is for A, A, B, B phase

※2: Make sure that, Max response revolution should be lower than or equal to max. allowable revolution when selecting the resolution.

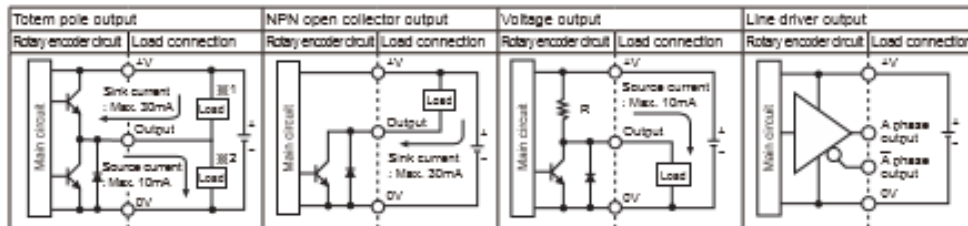
$$(\text{Max. response revolution/rpm}) = \frac{\text{Max. response frequency}}{\text{Resolution}} \times 60 (\text{sec})$$

※Environment resistance is rated at no freezing or condensation.



# Incremental ø40mm Shaft/Hollow Shaft/Built-in type

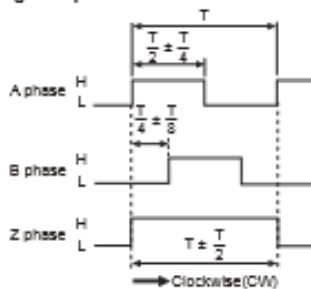
## Control output diagram



- Totem pole output type can be used for NPN open collector output type(※1) or Voltage output type(※2).
- All output circuits of A, B, Z phase are the same. (Line driver output is A,  $\bar{A}$ , B,  $\bar{B}$ , Z,  $\bar{Z}$ )

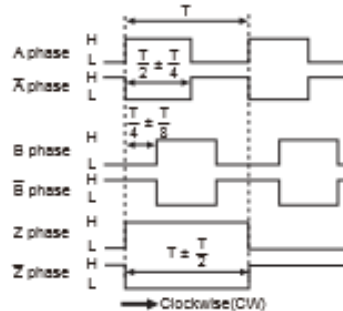
## Output waveform

- Totem pole output / NPN open collector output / Voltage output



※Z phase output is option.  
 ※CW: Right turn as from the shaft.

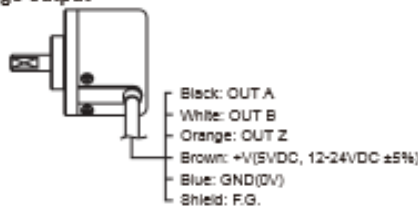
- Line driver output



## Connections

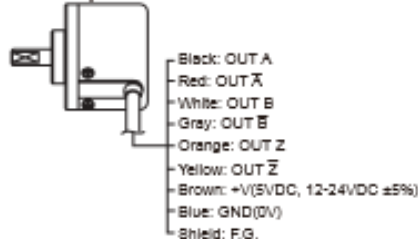
### Connector cable type

- Totem pole output / NPN open collector output / Voltage output



※Unused wires must be insulated.  
 ※The metal case and shield wire of encoder should be grounded(F.G.).

- Line driver output



### Connector cable type

- Totem pole output / NPN open collector output / Voltage output
- Line driver output



Totem pole output/ NPN open collector output/ Voltage output			Line driver output		
Pin No	Function	Cable color	Pin No	Function	Cable color
①	OUT A	Black	①	OUT A	Black
②	OUT B	White	②	OUT $\bar{A}$	Red
③	OUT Z	Orange	③	+V	Brown
④	+V	Brown	④	GND	Blue
⑤	GND	Blue	⑤	OUT B	White
⑥	F.G.	Shield	⑥	OUT $\bar{B}$	Gray
			⑦	OUT Z	Orange
			⑧	OUT $\bar{Z}$	Yellow
			⑨	F.G.	Shield

※F.G.(Field Ground): It should be grounded separately.

- (A) Photo switch sensor
- (B) Photo eye sensor
- (C) One/Two sensor
- (D) Proximity sensor
- (E) Pressure sensor
- (F) Rotary encoder
- (G) Connector/Socket
- (H) Temp. controller
- (I) Power controller
- (J) Counter
- (K) Timer
- (L) Encoder motor
- (M) Encoder/ Pulse motor
- (N) Display unit
- (O) Sensor controller
- (P) Bulb/Ink media power supply
- (Q) Blower motor & Drive Controller
- (R) Graphical Logic panel
- (S) Photo network device
- (T) Software
- (U) Other

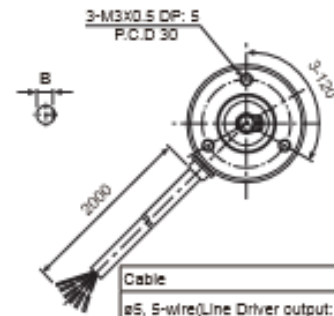
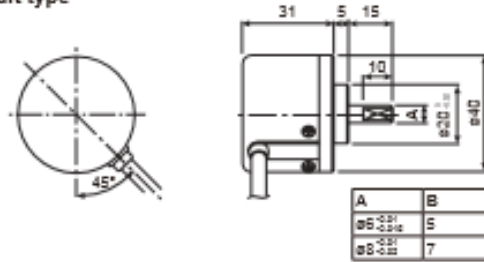


## E40 Series

### ■ Dimensions

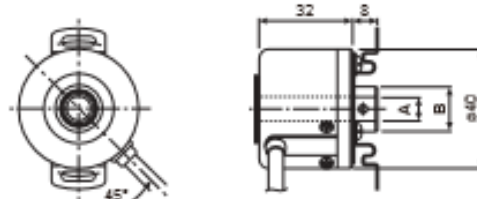
- ⊙ Cable type
- Shaft type

(unit: mm)

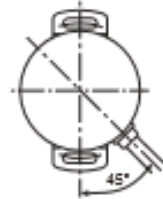


Cable  
φ5, 5-wire(Line Driver output: 8-wire),  
Length:2000, Shield cable

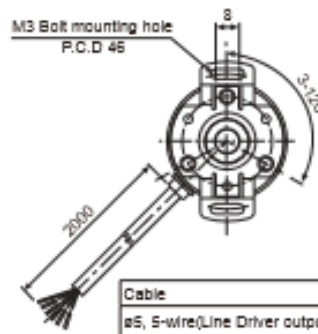
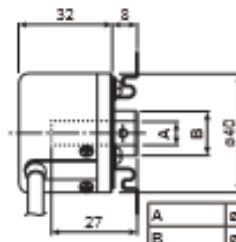
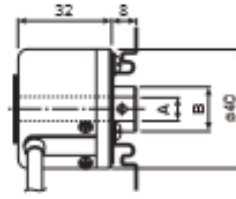
### ● Hollow shaft / Hollow shaft built-in type



E40H

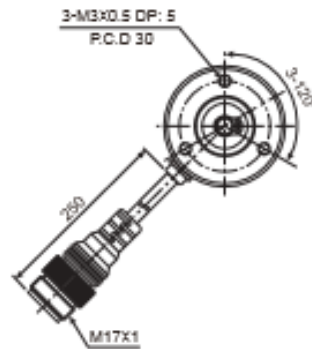


E40HB



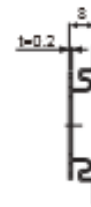
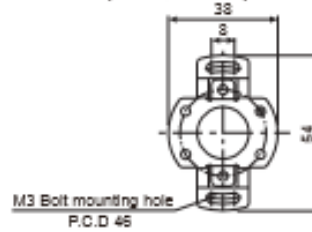
Cable  
φ5, 5-wire(Line Driver output: 8-wire),  
Length: 2000, Shield cable

### ⊙ Connector cable type



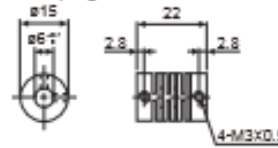
※Connector cable is customizable and refer to the G-10 for specifications.

### ● Bracket(E40H, E40HB)



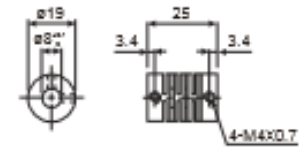
### ● Coupling(E40S)

#### ● φ6 Coupling



- Parallel misalignment : Max. 0.25mm
- Angular misalignment: Max. 5°
- End-play: Max. 0.2mm

#### ● φ8 Coupling



- ※For parallel misalignment, angular misalignment, end-play terms, refer to the F-78 page.
- ※For flexible coupling(ERB Series) information, refer to the F-71 page.

## L298 Dual H-Bridge Motor Driver

Double H driver module uses ST L298N dual full-bridge driver. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors.



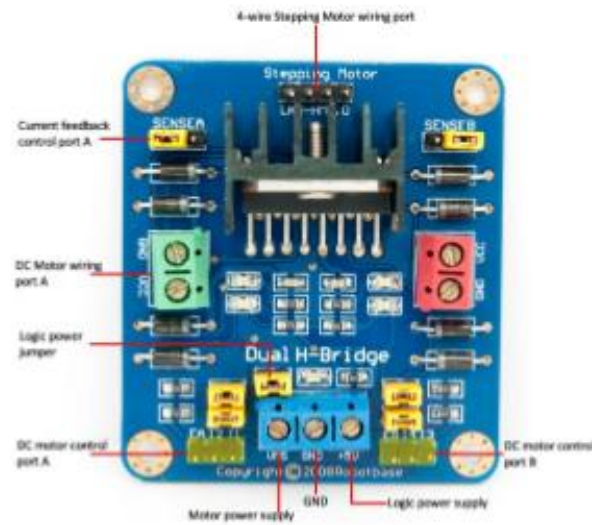
## Features

- Light weight, small dimension
- Super driver capacity
- FWD protection
- Heavy load Heat sink
- Power selection switch
- 4 pull up resistor switch
- 2 DC motor/ 4 coil dual phrase stepper motor output
- Motor direction indication LED
- 4 standard mouting holes

## Specifications

- Driver: L298
- Driver power supply: +5V~+46V
- Driver peak current: 2A
- Logic power output Vss: +5~+7V (internal supply +5V)
- Logic current: 0~36mA
- Controlling level: Low -0.3V~1.5V, high: 2.3V~Vss
- Enable signal level: Low -0.3V~1.5V, high: 2.3V~Vss
- Max drive power: 25W (Temperature 75 ℃)
- Working temperature: -25℃~+130℃
- Dimension: 60mm\*54mm
- Driver weight: ~48g

## Hardware Installation



Double H driver module can drive two DC motors at the same time. Port A is completely symmetrical as port B on the board.

DC motor input port A has three pins, I1, I2 and EA. I1 and I2 are digital ports which are used to control the direction of motor, EA is connecting with PWM port of control board to control the speed of motor.

If I1=1 and I2=0, the motor rotates clockwise.

If I1=0 and I2=1, it rotates anticlockwise.

If I1=I2, it stops rotating.

EA	I1	I2	Motor A status
0	0	1	Clockwise rotation
0	1	0	Anticlockwise rotation

Normally we do not need to connect wire to supply logic power.

Only if the motor power supply is +5V~+7V or +18V~+46V we have to take off the logic power jumper and connect 5v to supply logic power.

We can control 4-wire stepping Motor the same way as two DC motors with the signals from EA, I1, I2 and EB, I3, I4.