

**PEMODELAN EKSPERIMEN DAN PENGONTROLAN
KETINGGIAN AIR PADA TANGKI**



SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan pendidikan diploma empat (D-4) Program Studi Teknik Manufaktur
Jurusan Teknik Mesin
Politeknik Negeri Ujung Pandang

MUH. AKBAR
KAHARUDDIN

444 16 039
444 16 044

**PROGRAM STUDI D4 TEKNIK MANUFAKTUR
JURUSAN TEKNIK MESIN
POLITEKNIK NEGERI UJUNG PANDANG
MAKASSAR**

2018

HALAMAN PENGESAHAN

Skripsi degan judul **Pemodelan Eksperimen dan Pengontrolan Ketinggian Air Pada Tangki** Oleh Muh. Akbar (444 16 039) dan Kaharuddin (444 16 044) telah diterima dan disahkan sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains Terapan pada Program studi D4 Teknik Manufaktur Jurusan Teknik Mesin Politeknik Negeri Ujung Pandang.

Makassar, 3 September 2018

Pembimbing I,



Ir. Lewi, M.T.
NIP. 19650913 199103 1 006

Pembimbing II,



Ir. Simon Ka'ka, M.T.
NIP. 19590913 198803 1 001

Mengetahui,
Ketua Program Studi D4
Teknik Manufaktur









Ir. Abdul Salam, M.T.
NIP.19601224 199103 1 001

HALAMAN PENERIMAAN

Pada hari ini, Senin tanggal 3 September 2018, Tim Penguji Seminar Skripsi telah menerima dengan baik hasil seminar skripsi oleh mahasiswa: Muh. Akbar NIM 444 16 039 dan Kaharuddin NIM 444 16 044 dengan judul **Pemodelan Experimen dan Pengontrolan Ketinggian Air Pada Tangki**

Makassar, 3 September 2018

Tim Seminar Skripsi:

- | | | |
|---|---------------|---|
| 1. Dr.Eng. Akhmad Taufik, S.T.,M.T | Ketua | () |
| 2. Abdul Rahman, S.T., M.T. | Sekretaris | () |
| 3. Muhammad Iswar, S.ST.,M.T. | Anggota | () |
| 4. Dr.Eng. Abdul Kadir Muhammad, S.T.,M.Eng | Anggota | () |
| 5. Ir. Lewi., M.T | Pembimbing I | () |
| 6. Ir. Simon Ka'Ka.,M.T | Pembimbing II | () |

KATA PENGANTAR

Puji dan syukur penulis ucapkan kehadiran Tuhan Yang Maha Esa atas rahmat dan berkahnya penulis dapat menyelesaikan Tugas Akhir ini dengan judul **“Pemodelan Eksperimen dan Pengontrolan Ketinggian Air pada tangki”**. Tugas akhir ini disusun sebagai salah satu syarat guna memperoleh gelar Sarjana Sains Terapan di Politeknik Negeri Ujung Pandang Jurusan Teknik Mesin Program Studi D4 Teknik Manufaktur.

Dalam penyelesaian Tugas Akhir ini, penulis telah banyak mendapat bimbingan, saran dan bantuan dari berbagai pihak. Untuk itu penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Dr. Ir. Hamzah Yusuf, M.S., selaku Direktur Politeknik Negeri Ujung Pandang.
2. Dr. Jamal, S.T., M.T., selaku Ketua Jurusan Teknik Mesin Politeknik Negeri Ujung Pandang.
3. Ir. Lewi, M.T. Selaku Ketua Program Studi Teknik Mekatronika Jurusan Teknik Mesin Politeknik Negeri Ujung Pandang yang selalu memberikan bimbingan kepada penulis dalam menyelesaikan Tugas Akhir ini.
4. Ir. Lewi, M.T. selaku pembimbing 1 dan Ir. Simon Ka'ka, M.T selaku pembimbing II yang dengan ikhlas memberikan bimbingan serta dukungan moril kepada penulis dalam menyelesaikan Tugas Akhir ini.
5. Dr.Eng. Abdul Kadir Muhammad, S.T.,M.Eng. selaku dosen program studi D-4 Teknik Mekatronika yang telah mendidik dan memberikan

semangat dan inspirasi sehingga laporan tugas akhir ini dapat dikerjakan dengan baik.

6. Imran Habriansyah, S.ST., M.T. yang telah banyak membantu dan mengarahkan penulis, khususnya dalam pemrograman sistem.
7. Segenap dosen dan staf program studi Teknik Mekatronika Politeknik Negeri Ujung Pandang yang telah banyak membantu penulis.
8. Seluruh dosen dan teknisi yang telah mendidik dan mengajar serta memberikan ilmu kepada penulis selama penulis menuntut ilmu di Program Studi D4 Teknik Mekatronika Jurusan Teknik Mesin Politeknik Negeri Ujung Pandang.
9. Saudara(i) di Program Studi D4 Teknik mekatronika Jurusan Teknik Mesin Politeknik Negeri Ujung Pandang angkatan 2016 yang telah membantu dalam penyelesaian Tugas Akhir ini.
10. Semua pihak yang tidak sempat disebutkan satu persatu yang telah membantu penyelesaian Tugas Akhir ini.

Penulis menyadari bahwa skripsi ini belum sempurna. Oleh karena itu, penulis mengharapkan kritikan dan saran yang bersifat membangun demi kesempurnaan laporan tugas akhir ini, semoga apa yang penulis telah lakukan dapat memberi manfaat dan bernilai ibadah oleh Tuhan Yang Maha Esa, dan memberikan perlindungan kepada kita semua Amiiin...

Makassar, September 2018

Penulis

DAFTAR ISI

SAMPUL DEPAN	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENERIMAAN PANITIA UJIAN	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
DAFTAR LAMPIRAN	xii
SURAT PERNYATAAN	xiii
SURAT PERNYATAAN	xiv
RINGKASAN	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Ruang Lingkup Penelitian	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA	4
2.1 Kontrol Level Air	4
2.2 Pemodelan Sistem	4
2.3 Sistem Kontrol	6
2.3.1 Mikrokontroler	7
2.3.2 Kontrol <i>Proportional, Integral, Derivatif</i> (PID)	8
2.3.2.1. Kontrol Proporsional (P) untuk Motor DC	9
2.3.2.2. Kontrol Integral (I) untuk Motor DC	11

2.3.2.3. Kontrol Derivatif (D) untuk Motor DC	12
2.4 Pompa Air.....	13
2.4.1. Klasifikasi Pompa	14
2.4.2. Daya Pompa.....	14
2.4.3. Aliran Fluida Dinamis	15
2.5 Sensor	16
2.5.1. Sensor Ultrasonik.....	16
2.5.2. Pengertian Sensor Ultrasonik.....	17
2.5.3 Prinsip Kerja Penerima Ultrasonik (<i>Transmitter</i>).....	20
2.5.4 Prinsip Kerja Penerima Ultrasonik (<i>Receiver</i>).....	21
2.6 Driver Motor.....	22
2.7 Bahasa Pemrograman Matlab.....	23
2.7.1. Defenisi Pemrograman Matlab	23
2.7.2. Matriks.....	23
2.7.3. M-File	24
BAB III METODE PENELITIAN	26
3.1 Waktu dan Tempat Pelaksanaan	26
3.2 Alat dan Bahan yang Digunakan	26
3.3 Metode Penelitian/Perancangan	27
3.3.1. Studi Literatur	27
3.3.2. Perancangan <i>Hardware</i> dan <i>Software</i>	27
3.3.3. Pengujian dan Pengambilan Data.....	28
3.3.4. Pengolahan Data.....	29
3.4 Prosedur Perancangan	30
3.4.1 Diagram Blok.....	30
3.4.2 Penentuan Parameter Model Sistem Melalui Pengujian	30
3.4.2. Diagram Alir	30

3.5	Prosedur Penelitian	32
3.6.	Desain <i>Water Level Control</i> pada Pemodelan Experimen dan Pengontrolan Ketinggian Air pada Tangki.....	32
BAB IV HASIL DAN PEMBAHASAN		34
4.1	Hasil Perancangan	34
4.1.1	Hasil dan Perancangan Perangkat Keras	34
4.1.2	Hasil dan Perancangan Perangkat Lunak	37
4.1.3	Proses Kerja Alat	38
4.1.4	Cara Pengoperasian Alat	39
4.2	Hasil Pengujian dan Pengambilan Data	41
4.2.1	Pemodelan <i>Steady State</i>	41
4.2.2	<i>Step Response</i>	44
4.3	Analisa Hasil Pengujian	47
BAB V PENUTUP		49
5.1	Kesimpulan	49
5.2	Saran	49
DAFTAR PUSTAKA		



DAFTAR TABEL

Tabel 3.1 Alat dan Bahan Perancangan	26
Tabel 3.2 Komponen dan Spesifikasi	28
Tabel 4.1 Hubungan Tegangan Input dan Output	42



DAFTAR GAMBAR

Gambar 2.1 Langkah-Langkah Penegembangan Model.....	5
Gambar 2.2 Arduino AT Mega	8
Gambar 2.3 Kontrol Loop Tertutup.....	9
Gambar 2.4 Diagram Kontrol P.....	9
Gambar 2.5 Kontrol P pada Motor DC.....	10
Gambar 2.6 Kontrol I pada Motor DC	11
Gambar 2.7 Kontrol PI pada Motor DC.....	12
Gambar 2.8 Diagram Kontrol D.....	13
Gambar 2.9 Kontrol PD pada Motor DC	13
Gambar 2.10 Bentuk Fisik Pompa Air.....	14
Gambar 2.11 Prinsip Kerja Sensor Ultrasonik.....	18
Gambar 2.12 Prinsip Pemantulan Sensor Ultrasonik	20
Gambar 2.13 Pemancar Ultrasonic Transmitter	20
Gambar 2.14 Sensor Ultrasonic HC-SR04.....	22
Gambar 2.15 Driver Motor BTS7960.....	23
Gambar 3.1 Perangkat keras <i>Interface</i>	27
Gambar 3.2 Diagram Blok Sistem Pemodelan Experimen dan Pengontrolan Ketinggian Air pada Tangki.....	30
Gambar 3.3 Diagram Alir Pemodelan Experimen dan Pengontrolan Ketinggian Air pada Tangki.....	31
Gambar 3.4 Alat Sistem Pemodelan Experimen dan Pengontrolan Ketinggian Air pada Tangki.....	32

Gambar 3.5 Alat Kontrol Ketinggian Air dengan Perangkat <i>Interface</i>	33
Gambar 4.1 Mekanik Kontrol Level Air.....	35
Gambar 4.2 Layout Rangkaian Kontrol Level Air	36
Gambar 4.3 Rangkaian Kontrol Level Air.....	36
Gambar 4.4 Perangkat Keras <i>Interface</i>	37
Gambar 4.5 Listing Program Arduino Tegangan Awal dan Step	37
Gambar 4.6 Listing Program Arduino Pengaturan Nilai PID	38
Gambar 4.7 Cara Pengoperasian Alat.....	40
Gambar 4.8 Hubungan Tegangan Input dan Output.....	43
Gambar 4.9 Hubungan Waktu dan Level pada Tahap Step Respon	44
Gambar 4.10 Tampilan Trending Level dan Tegangan pada Matlab	44
Gambar 4.11 Pemodelan Berdasarkan Eksperimen.....	45
Gambar 4.12 Persamaan Transfer Fungsi pada Simulink.....	45
Gambar 4.13 Hasil dari Simulasi Matlab.....	46
Gambar 4.14 Pengaturan Nilai PID pada <i>Simulink</i>	47
Gambar 4.15 Hasil Grafik pada Simulink.....	47

DAFTAR LAMPIRAN

Lampiran 1 Listing Program	50
Lampiran 2 Hasil Pengujian dan Pengolahan Data	65
Lampiran 3 Datasheet Arduino dan Sensor Ultrasonik	76
Lampiran 4 Dokumentasi Pengujian Alat dan Pengambilan Data	87



SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Muh. Akbar

NIM : 444 16 039

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam Tugas Akhir ini yang berjudul “Pemodelan Eksperimen dan Pengontrolan Ketinggian Air pada tangki” merupakan gagasan dan hasil karya kami sendiri dengan arahan komisi pembimbing dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi atau instansi apapun.

Semua data dan informasi telah dinyatakan dengan jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya penulis lain telah disebutkan dalam naskah dan dicantumkan dalam Laporan Tugas Akhir ini.

Jika pernyataan kami di atas tidak benar, kami siap menanggung resiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, September 2018



Muh. Akbar
444 16 039

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Kaharuddin

NIM : 444 16 044

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam Tugas Akhir ini yang berjudul “Pemodelan Eksperimen dan Pengontrolan Ketinggian Air pada tangki” merupakan gagasan dan hasil karya kami sendiri dengan arahan komisi pembimbing dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi atau instansi apapun.

Semua data dan informasi telah dinyatakan dengan jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya penulis lain telah disebutkan dalam naskah dan dicantumkan dalam Laporan Tugas Akhir ini.

Jika pernyataan kami di atas tidak benar, kami siap menanggung resiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, September 2018



Kaharuddin
444 16 044

RINGKASAN

Muh Akbar, Kaharuddin, **Pemodelan Eksperimen dan Pengontrolan Ketinggian Air Pada Tangki.** (Pembimbing I : Ir. Lewi, M.T. dan Pembimbing II: Ir. Simon Ka'ka, M.T.).

Pemodelan Eksperimen dan Pengontrolan Ketinggian Air Pada Tangki merupakan sebuah sistem yang bekerja berdasarkan dengan variabel nilai tegangan awal dan tegangan step yang dapat diatur pada perangkat *Interface*, serta level pada tangki penampungan air dapat dikontrol dengan mengatur nilai *set point* pada perangkat *Interface*. Sistem *water level control* ini terdiri dari Perangkat *Interface*, Sensor Ultrasonik, Motor DC, dan Pompa Air. Mikrokontroler yang terdapat pada perangkat *Interface* merupakan pusat kendali dari seluruh rangkaian, dimana mikrokontroler akan mengambil data yang dikirimkan sensor ultrasonik sehingga mikrokontroler dapat mengatur kecepatan pompa kemudian hasil pembacaan data sensor ultrasonik akan ditampilkan secara *real time* pada *Graphical User Interface* (GUI) matlab. Berdasarkan hasil penelitian tersebut dapat disimpulkan bahwa sistem *water level control* dapat dimodelkan dengan melakukan beberapa eksperimen untuk mendapatkan grafik yang sesuai, selanjutnya data tersebut disimulasikan pada pada matlab dengan mengatur nilai K_p , K_i dan K_d .

Kata Kunci : *Interface*, *Graphical User Interface* (GUI), Motor DC, Sensor Ultrasonik

BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi elektronika saat ini mengalami peningkatan yang sangat pesat. Perkembangan ini cenderung mengarah pada peningkatan optimalisasi kecepatan kerja dan minimalisasi. Artinya peralatan dan komponen elektronika diupayakan menggunakan materi dan ukuran yang semakin kecil tetapi mempunyai kemampuan kecepatan maupun kerja yang lebih tinggi.

Sistem kontrol level air merupakan sistem yang digunakan untuk menjamin kontinuitas persediaan air dalam sebuah tangki penampungan yang akan digunakan untuk proses industri atau dalam kehidupan sehari-hari. Misalnya, pada industri minuman, industri pengolahan air bersih, Pembangkit Listrik Tenaga Uap (PLTU), industri kimia, dan lain-lain. Selain itu sistem kontrol ini juga dapat digunakan untuk memonitoring proses industri yang dikendalikan dari jauh, sehingga bisa menghemat biaya, waktu dan tenaga.

Untuk menjaga level ketinggian air pada tangki maka digunakan sebuah alat kontrol, yaitu mikrokontroler untuk mengatur kecepatan pompa sesuai dengan yang diinginkan. Selain itu juga dibutuhkan sebuah sensor yang berfungsi untuk mendeteksi setiap penurunan level air dan ditindaklanjuti melalui sistem kontrol tersebut.

Penelitian sebelumnya mengenai sistem kontrol level, dilakukan pada tahun 2016 dengan judul “Rancang Bangun Kontrol Level Air Berbasis Mikrokontroler Menggunakan Sensor Ultrasonik”. Pada tahun 2017 dengan

judul “Pengembangan Kontrol Level dan Temperatur Air Berbasis Mikrokontroler”, mereka menambahkan sebuah kontrol PID, dan menambah satu sistem yaitu sistem kontrol temperatur. Pada kesempatan ini kami akan memodelkan sistem tersebut, dengan judul “Pemodelan Eksperimen dan Pengontrolan Ketinggian Air Pada Tangki”

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dipaparkan sebelumnya dapat dirumuskan sebagai berikut:

- 1) Bagaimana membuat perangkat *interface* dari sistem pengontrol ketinggian air?
- 2) Bagaimana cara menampilkan ketinggian air pada tangki secara *real time* pada komputer

1.3 Ruang Lingkup Penelitian

Dalam pembuatan tugas akhir ini dibatasi beberapa hal sebagai berikut :

- 1) Sensor yang digunakan untuk mendeteksi level air adalah sensor ultrasonik.
- 2) Arduino Mega sebagai alat untuk memproses data dan pengatur dari seluruh kegiatan sistem yang dibuat.
- 3) Ketinggian air dapat ditentukan antara 0% – 100%.
- 4) Ketinggian air dapat ditampilkan dalam bentuk GUI (*Grafical User Interface*) pada komputer.

1.4 Tujuan Penelitian

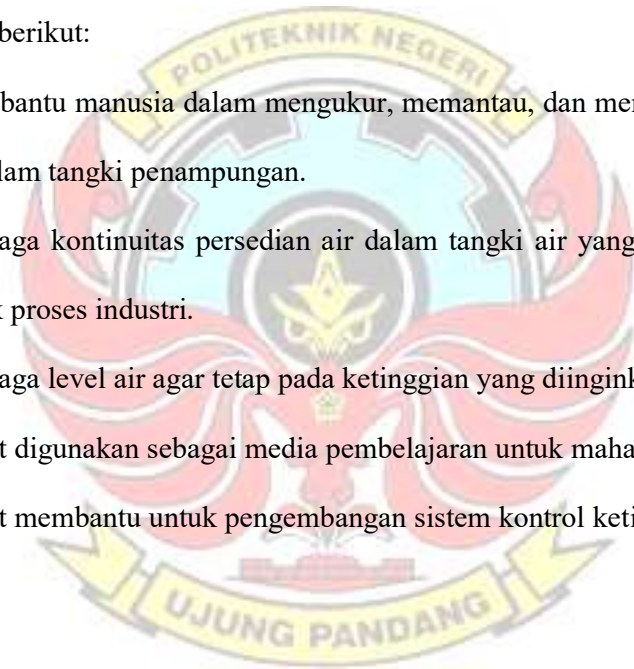
Adapun tujuan dari proyek tugas akhir ini adalah sebagai berikut:

- 1) Untuk membuat perangkat *interface* dari sistem pengontrol ketinggian air?
- 2) Untuk menampilkan ketinggian air pada tangka secara *real time* pada komputer.

1.5 Manfaat Penelitian

Adapun manfaat yang akan diperoleh dari proyek tugas akhir ini adalah sebagai berikut:

- 1) Membantu manusia dalam mengukur, memantau, dan mengontrol level air di dalam tangki penampungan.
- 2) Menjaga kontinuitas persedian air dalam tangki air yang akan digunakan untuk proses industri.
- 3) Menjaga level air agar tetap pada ketinggian yang diinginkan.
- 4) Dapat digunakan sebagai media pembelajaran untuk mahasiswa.
- 5) Dapat membantu untuk pengembangan sistem kontrol ketinggian air.



BAB II TINJAUAN PUSTAKA

2.1 Kontrol Level Air

“*Water Level Control (WLC)*” merupakan sebuah alat yang digunakan untuk mengidentifikasi ketinggian air yang berada di dalam penampungan dengan memanfaatkan sensor sebagai pendeteksi ketinggian dari air dan memiliki aktuator seperti motor sebagai pemompa air. Sistem ini banyak digunakan di dalam dunia industri maupun di dalam kehidupan sehari-hari, karena sederhana dan perawatannya mudah.

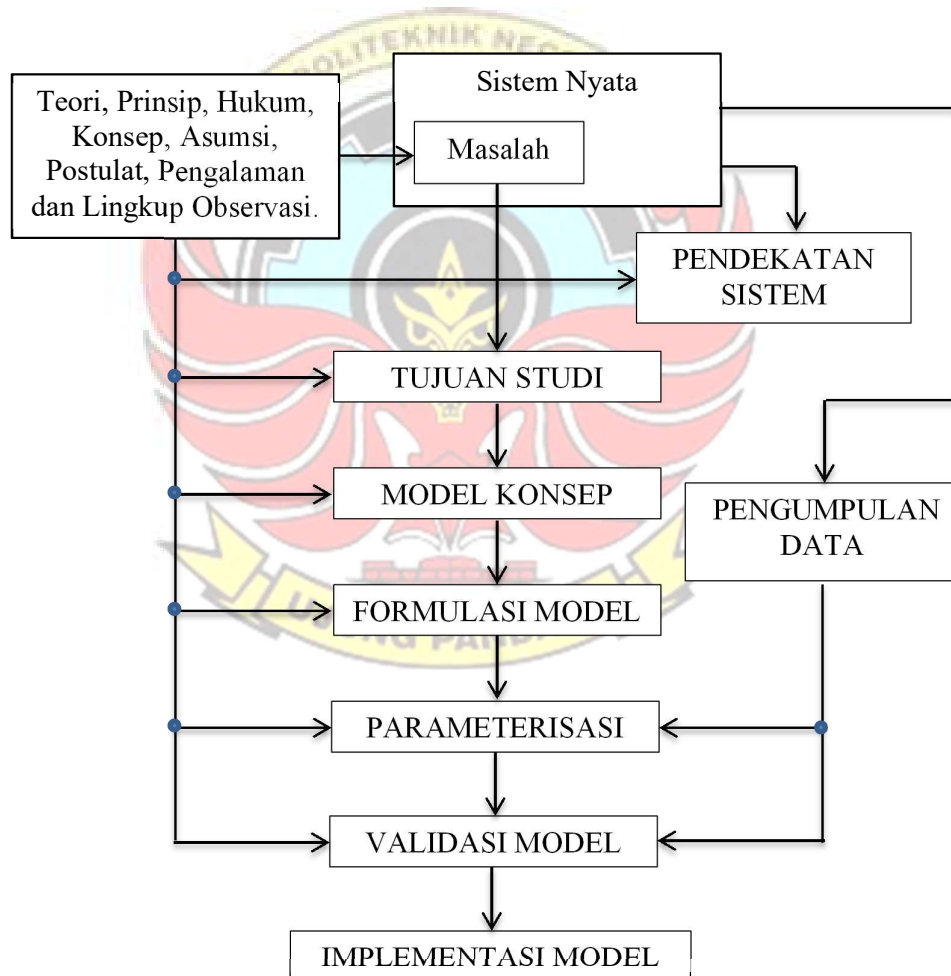
2.2 Pemodelan Sistem

Menurut Sridadi (2009), model adalah suatu representasi atau formalisasi dalam bahasa tertentu dari suatu sistem nyata (realitas). Model berisi informasi tentang suatu sistem yang dibuat dengan tujuan untuk mempelajari perilaku sistem yang sebenarnya. Kebanyakan model untuk simulasi biasanya dalam bentuk matematika. Sebelum dapat mensimulasikan sistem maka terlebih dahulu harus dikembangkan model dari sistem yang bersangkutan. Tujuan dari studi pemodelan adalah menentukan informasi (variabel dan parameter) yang dianggap penting untuk dikumpulkan, sehingga tidak ada model yang unik.

Menurut Santoso (2002), pemodelan (*Modelling*) berarti mengubah sistem dalam dunia nyata (*real world*) menjadi bentuk persamaan matematika. *Modelling* juga dapat diartikan sebagai usaha menirukan kelakuan proses “*real world system*” dalam usaha untuk memahaminya. Hal tersebut dilakukan dengan menyusun hubungan-hubungan fisik dari sistem sesungguhnya dengan

menggunakan hukum-hukum ilmu alam (fisika dan atau kimia). Tanggapan sistem dinamik dapat diperoleh, jika persamaan differensial tersebut dicari solusinya. Banyak metode untuk mencari solusi persamaan differensial sistem, misalnya dengan menggunakan metode Transformasi Laplace atau metode ruang keadaan (*state space*).

Menurut Sridadi (2009), langkah-langkah pengembangan model seperti pada gambar 1.



Gambar 2.1 Langkah-langkah pengembangan model

Penggunaan model matematika memungkinkan kita dapat mengembangkan teori Kontrol yang menyatu. Untuk mengembangkan suatu sistem maka sebelum diimplementasikan pertama harus memformulasikan model matematikanya sesuai dengan komponen dan tujuan kinerja sistem, kemudian menterjemahkan model tersebut ke dalam program komputer untuk mensimulasikan sistem tersebut. Dari simulasi tersebut, kita dapat menentukan ukuran komponen yang diperlukan.

2.3 Sistem Kontrol

Sistem kontrol adalah proses pengaturan pengendalian terhadap satu atau beberapa besaran (*variable, parameter*) sehingga berada pada suatu kisaran tertentu. Gabungan kerja dari berbagai alat-alat kontrol dalam proses produksi dinamakan sistem pengontrolan proses (*process control sistem*), sedangkan semua peralatan yang membentuk sistem pengontrolan disebut pengontrolan instrumentasi proses (*process control instrumentation*).

Menurut Curtis D. Johnson (1989), mode alat kontrol terbagi atas dua macam, yaitu mode alat kontrol diskontinyu dan mode alat kontrol kontinyu. Mode alat kontrol diskontinyu terbagi atas 2 macam, yaitu mode dua posisi dan mode multi posisi. Mode dua posisi ini hanya bekerja pada dua posisi, yaitu *on* dan *off*. Kerja kontroler *on-off* banyak digunakan pada aksi pengontrolan yang sederhana karena harganya murah dan mode alat kontrol ini merupakan yang paling dasar. Mode multi posisi adalah mode alat kontrol dua posisi yang diberikan beberapa pengaturan lanjutan (*intermediate setting*).

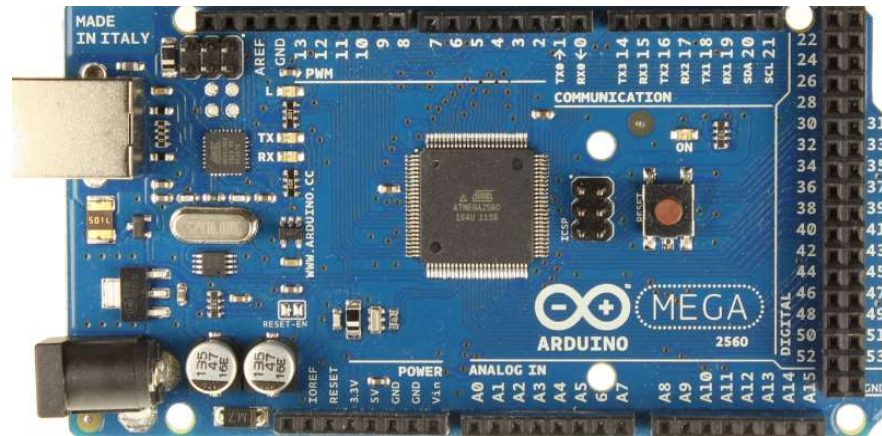
2.3.1 Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah *chip* berupa IC (*Integrated Circuit*) yang dapat menerima sinyal input, mengolahnya dan memberikan sinyal output sesuai dengan program yang diisikan ke dalamnya. Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data.

Mikrokontroler pada dasarnya adalah komputer dalam satu *chip*, yang di dalamnya terdapat mikroprosesor, memori, jalur *Input/Output* (I/O) dan perangkat pelengkap lainnya. Kecepatan pengolahan data pada mikrokontroler lebih rendah jika dibandingkan dengan PC. Pada PC kecepatan mikroprosesor yang digunakan saat ini telah mencapai orde GHz, sedangkan kecepatan operasi mikrokontroler pada umumnya berkisar antara 1 – 16 MHz. Begitu juga kapasitas RAM dan ROM pada PC yang bisa mencapai orde *Gbyte*, dibandingkan dengan mikrokontroler yang hanya berkisar pada orde *byte/Kbyte*.

Salah satu mikrokontroler yang sering digunakan adalah Arduino merupakan rangkaian elektronik yang bersifat *open source*, serta memiliki perangkat keras dan lunak yang mudah untuk digunakan. Arduino dapat mengenali lingkungan sekitarnya melalui berbagai jenis sensor dan dapat mengendalikan lampu, motor, dan berbagai jenis

aktuator lainnya. Arduino mempunyai banyak jenis, di antaranya Arduino Uno, Arduino Mega 2560, Arduino Fio, dan lainnya.

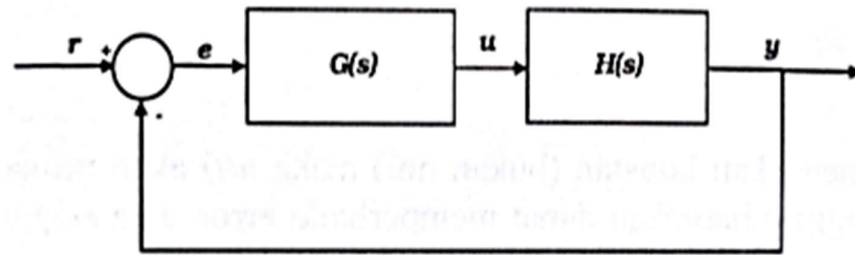


Gambar 2.2 Arduino AT Mega

Arduino adalah sebuah *board mikrokontroller* yang berbasis ATmega328. Arduino memiliki 14 pin input/output yang mana 6 pin dapat digunakan sebagai output PWM, 6 *analog input*, *crystal osilator* 16 MHz, koneksi USB, *jack power*, kepala ICSP, dan tombol *reset*. Arduino mampu *men-support* mikrokontroller dan dapat dikoneksikan dengan komputer menggunakan kabel USB. (Feri Djuandi, 2011)

2.3.2 Kontrol *Proporsional, Integral, Derivatif* (PID)

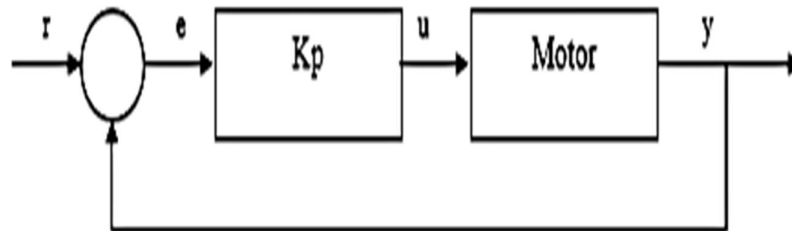
Kita akan menggambarkan bentuk pernyataan standar dalam sistem kontrol seperti pada gambar dibawah berikut. Dalam gambar, r adalah input, e adalah *error*, u adalah sinyal *output* kontroler, $G(s)$ adalah kontroler, $H(s)$ adalah dinamik robot, dan y adalah output. Sekarang permasalahannya adalah bagaimana $G(s)$ didisain.



Gambar 2.3 Kontrol Loop Tertutup

2.3.2.1 Kontrol Proporsional (P) untuk motor DC

Kontroler adalah kontrol P jika $G(s) = k$, dengan k adalah konstanta jika, $u = G(s).e$ maka Persamaan output kontroler $u = k_p.e$ Kontrol P untuk sebuah motor dapat diilustrasikan sebagai berikut.

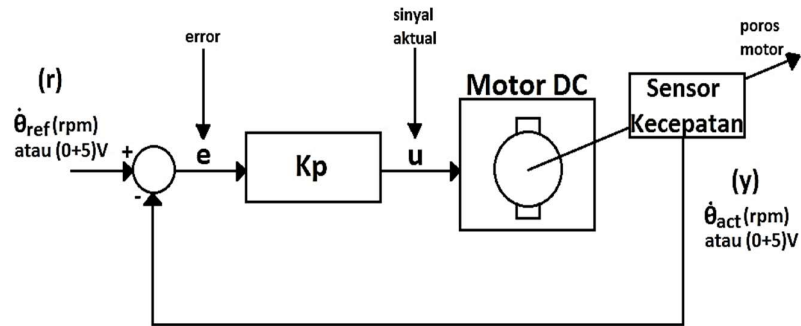


Gambar 2.4 Diagram Kontrol P

Dengan K_p adalah konstanta proporsional. K_p berlaku sebagai *Gain* (Penguat) saja tanpa memberikan efek dinamik pada kinerja kontroler.

Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi aplikasi dasar sederhana, kontrol P ini cukup mampu

untuk konvergensi meskipun *error* keadaan tenangnya (*steady state*) relatif besar. Sebagai contoh, kita akan mengontrol kecepatan putar sebuah motor DC dengan menggunakan kontrol P, ilustrasinya adalah sebagai berikut:



Gambar 2.5 Kontrol P pada Motor DC

Dalam fungsi t maka output kontroler $u(t)$ dapat ditulis:

$$u(t) = Kp. (\theta_{ref}(t) - \theta_{act}(t)) \dots\dots\dots(2.1)$$

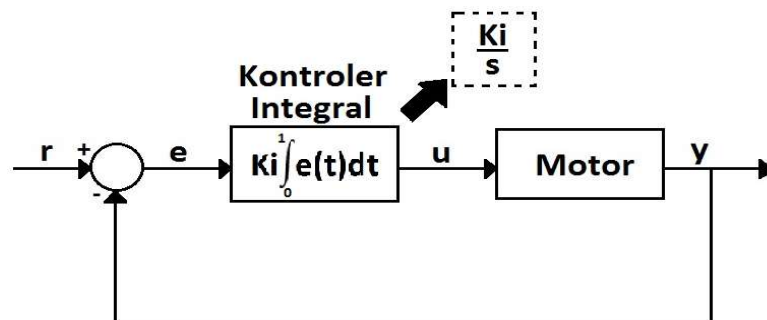
Nilai output kontroler tergantung kepada perwakilan antara *error*, yakni kecepatan referensi dikurangi kecepatan aktual, dengan lebih kecil daripada kecepatan referensi. Jika *error* negatif berarti kecepatan aktual lebih besar daripada kecepatan referensi. Jika kecepatan aktual sama dengan kecepatan referensi maka sinyal output akan menjadi nol karena error nol, dari segi rangkaian, sinyal output ini akan menyebabkan motor tidak mendapat sinyal aktuasi lagi. Akibatnya poros motor akan berhenti berputar. Begitu putaran poros mulai berkurang maka sensor akan mendeteksi bahwa kecepatan output tidak lagi sama

dengan kecepatan referensi. Kecepatan aktual akan lebih kecil dari kecepatan referensi. Artinya error tidak lagi nol. Akibatnya, kontroler akan mulai lagi mengirim sinyal aktuasi $u(t)$ positif sehingga motor kembali menambah kecepatannya.

Demikian hal ini berulang seterusnya sehingga error pada kontrol P ini tidak dapat mempertahankan error selalu nol atau dengan kata lain dalam kondisi tetap (*steady state*), error pada kontrol P tidak bisa nol. Hal ini dikenal sebagai *steady state error* (e_{ss}).

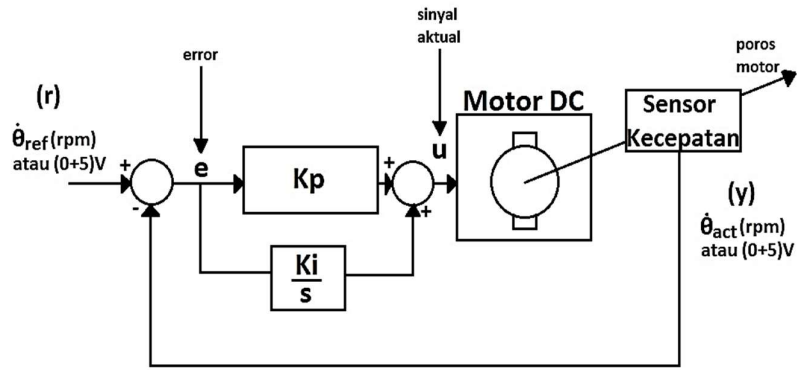
2.3.2.2 Kontrol Integral (I) untuk motor DC

Fungsi dasar dari kontrol I adalah menurunkan *steady state error*. Kontrol I jarang digunakan sendirian dalam aplikasi. Biasanya selalu dikombinasikan dengan kontrol P untuk memperbaiki respon guna mencapai error minimum. Parameter kontrol I dapat diilustrasikan dalam diagram kontrol motor sebagai berikut.



Gambar 2.6 Diagram Kontrol I

Gabungan kontrol I dengan kontrol P untuk sebuah motor DC dapat digambarkan sebagai berikut:



Gambar 2.7 Kontrol PI pada motor DC

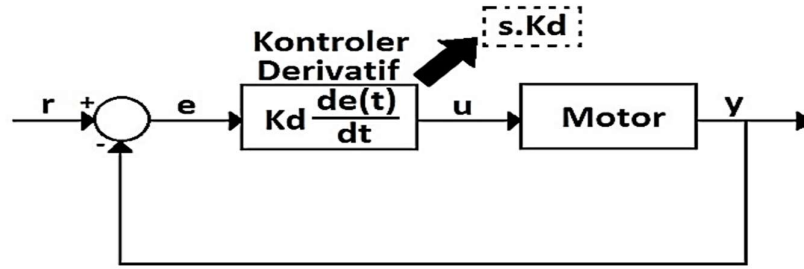
Persamaan output kontroler diatas adalah:

$$u(t) = Kp \cdot (\theta_{ref}(t) - \theta_{act}(t)) + Ki \int_0^t (\theta_{ref}(t) - \theta_{act}(t)) dt \dots \dots \dots (2.2)$$

2.3.2.3 Kontrol Derifatif (D) untuk motor DC

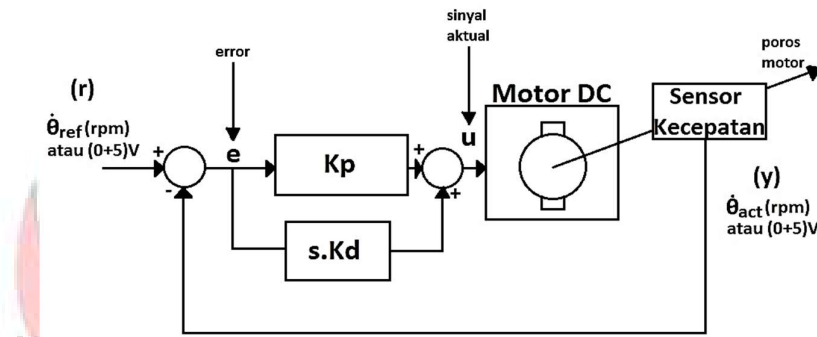
Parameter D bekerja dalam konteks rate/kecepatan dari error sehingga dapat mengurangi efek overshoot (respon berlebihan) dalam menuju steady state. Dengan kata lain, seolah olah kontrol D mampu memprediksi error yang akan terjadi sebagai efek dari kecepatan error yang dihitung sebelumnya.

Parameter kontrol D dapat diilustrasikan dalam diagram kontrol motor sebagai berikut:



Gambar 2.8 Diagram Kontrol D

Gabungan kontrol D dengan kontrol P untuk sebuah motor DC dapat digambarkan sebagai berikut:



Gambar 2.9 Kontrol PD pada motor DC

Sehingga persamaan output kontroler $u(t)$,

$$u(t) = Kp. (\theta_{ref}(t) - \theta_{act}(t)) + Kd. \frac{d(\theta_{ref}(t) - \theta_{act}(t))}{dt} \dots\dots\dots(2.3)$$

2.4 Pompa Air

Pompa adalah salah satu mesin fluida yang termasuk dalam golongan mesin kerja. Pompa berfungsi untuk mengalirkan fluida dari tempat yang rendah ke tempat yang lebih tinggi atau dari tempat yang bertekanan rendah ke tempat yang bertekanan lebih tinggi. Disamping itu juga pompa digunakan untuk memindahkan fluida dari tempat yang lebih tinggi ke tempat yang lebih rendah, misalnya pada sistem pemipaan yang panjang dan berkelok-kelok.



Gambar 2.10 Bentuk Fisik Pompa Air

2.4.1 Klasifikasi Pompa

Pompa dapat diklasifikasikan secara umum berdasarkan pemberian energi pada fluida kerjanya, yaitu:

- 1) Pompa Tekanan Statis (*Positive Displacement Pump*)
- 2) Pompa Tekanan Dinamis (*Dynamic Pressure Pump*)

2.4.2 Daya Pompa

Daya yang diberikan kepada pompa harus lebih besar dari daya akibat fluida dan akibat dari kerugian-kerugian yang terjadi. Daya pompa dapat dihitung dengan persamaan:

$$P_p = \frac{\rho \cdot Q \cdot H \cdot g}{\eta_p} \dots\dots\dots(2.4)$$

Dimana :

P_p = daya pompa [watt]

H = ketinggian [m]

ρ = massa jenis fluida [kg/m³]

Q = Debit Fluida [m^3/s]

η_p = efisiensi pompa [%]

g = gravitasi bumi [m/s^2]

2.4.3 Aliran Fluida Dinamis

Ada beberapa jenis aliran fluida. Lintasan yang ditempuh suatu fluida yang sedang bergerak disebut garis alir. Berikut ini beberapa jenis aliran fluida.

- 1) Aliran lurus atau *laminar* yaitu aliran fluida mulus. Lapisan-lapisan yang berseblahan meluncur satu sama lain dengan mulus. Pada aliran partikel fluida mengikuti lintasan yang mulus dan lintasan ini tidak saling bersilangan aliran laminar dijumpai pada air yang dialirkan melalui pipa atau selang
- 2) Aliran turbulence yaitu aliran yang ditandai dengan adanya lingkaran-lingkaran yang tak menentu dan menyerupai pusaran. Aliran turbulence sering dijumpai di sungai dan selokan.

Adapun formulasi laju aliran pada fluida dinamis:

$$Q = \frac{\Delta v}{\Delta t} = \frac{Av\Delta t}{\Delta t} = Av \dots \dots \dots (2.5)$$

Dimana:

Q = Debit aliran (m^3/s)

A = Luas penampang (m^2)

v = Laju aliran fluida (m/s)

Aliran fluida sering dinyatakan dalam debit aliran

$$Q = \frac{V}{t} \dots \dots \dots (2.6)$$

Dimana :

Q = Debit aliran (m³/s)

V = Volume (m³)

t = Selang waktu (s)

2.5 Sensor

Adapun sensor yang digunakan pada tugas akhir ini yaitu:

2.5.1 Sensor Ultrasonik

Penelitian tentang sensor ultrasonik telah banyak dilakukan oleh peneliti-peneliti sebelumnya di antaranya oleh Heru Rianto (2010) dalam penelitiannya yang berjudul Pengaman Parkir Mobil Menggunakan Sensor Ultrasonik Berbasis Mikrokontroler. Hasil yang diperoleh dari penelitian ini adalah berupa alat yang dapat mendeteksi adanya benda penghalang yang berada di sekitar mobil.

K.G. Suastika, M. Nawir, P. Yunus dalam Jurnal Pendidikan Fisika Indonesia (2013), Universitas Palangka Raya tentang Sensor Ultrasonik sebagai Alat Pengukur Kecepatan Aliran Udara Dalam Pipa. Penelitian yang dilakukan adalah penelitian tentang pengukuran kecepatan aliran udara dalam pipa menggunakan sensor ultrasonik dan gelombang ultrasonik yang digunakan pada penelitian ini adalah gelombang ultrasonik yang memiliki frekuensi kerja sebesar 300 kHz. Prinsip pengukuran yang digunakan dalam penelitian ini menggunakan metode

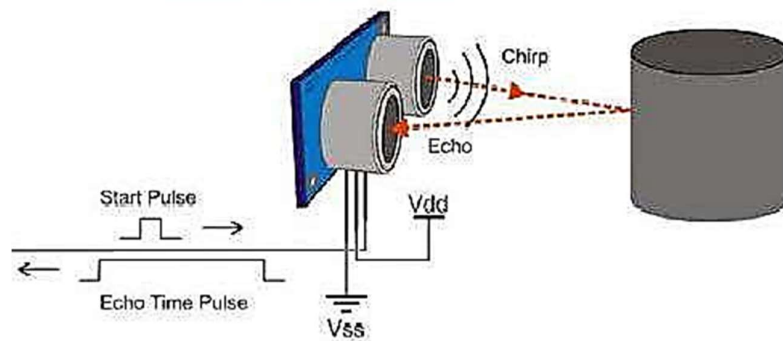
waktu tempuh gelombang ultrasonik (*time of flight*) dengan memanfaatkan perubahan karakteristik gelombang ultrasonik ketika melewati kondisi aliran udara yang berbeda yaitu *upstream* dan *downstream*. Selain itu, sebagai pembanding (tingkat akurasi) dalam penelitian ini digunakan alat pengukur kecepatan aliran udara standar yaitu *anemometer*. Dari hasil penelitian didapat bahwa tingkat akurasi sebesar 99% dan dengan korelasi sebesar 0,99 (korelasi sangat tinggi). Berdasarkan hasil tersebut, dapat disimpulkan bahwa sensor ultrasonik *valid* dan dapat digunakan sebagai perangkat pengukur kecepatan aliran udara dalam pipa. Berdasarkan penelitian tersebut, sensor ultrasonik baik digunakan untuk mengukur jarak suatu benda yang bersifat padat, cair, dan gas.

2.5.2 Pengertian Sensor Ultrasonik

Sensor adalah peralatan yang digunakan untuk merubah suatu besaran fisik menjadi besaran listrik sehingga dapat dianalisa dengan rangkain listrik tertentu (Budiarso: 2011). Sensor ultrasonik adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek tertentu di depannya, frekuensi kerjanya pada daerah di atas gelombang suara dari 40 KHz hingga 400 KHz. Sensor ultrasonik terdiri dari dari dua unit, yaitu unit pemancar dan unit penerima. Struktur unit pemancar dan penerima sangatlah sederhana, sebuah kristal *piezoelectric* dihubungkan dengan mekanik jangkar dan hanya dihubungkan dengan diafragma

penggetar. Tegangan bolak-balik yang memiliki frekuensi kerja 40 KHz – 400 KHz diberikan pada plat logam. Struktur atom dari kristal *piezoelectric* akan berkontraksi (mengikat), mengembang atau menyusut terhadap polaritas tegangan yang diberikan dan ini disebut dengan efek *piezoelectric*.

Kontraksi yang terjadi diteruskan ke diafragma penggetar sehingga terjadi gelombang ultrasonik yang dipancarkan ke udara (tempat sekitarnya). Pantulan gelombang ultrasonik akan terjadi bila ada objek tertentu dan pantulan gelombang ultrasonik akan diterima kembali oleh unit sensor penerima. Selanjutnya unit sensor penerima akan menyebabkan diafragma penggetar akan bergetar dan efek *piezoelectric* menghasilkan sebuah tegangan bolak-balik dengan frekuensi yang sama. Untuk lebih jelas tentang prinsip kerja dari sensor ultrasonik dapat dilihat prinsip dari sensor ultrasonic pada gambar berikut :



Gambar 2.11 Prinsip kerja sensor ultrasonik

Besar amplitudo sinyal elektrik yang dihasilkan sensor penerima tergantung dari jauh dekatnya objek yang dideteksi serta kualitas dari sensor pemancar dan sensor penerima. Proses yang dilakukan pada sensor ini menggunakan metode pantulan untuk menghitung jarak antara sensor dengan obyek sasaran. Jarak antara sensor tersebut dihitung dengan cara mengalikan setengah waktu yang digunakan oleh sinyal ultrasonik dalam perjalanannya dari rangkaian pengirim sampai diterima oleh rangkaian penerima, dengan kecepatan rambat dari sinyal ultrasonik tersebut pada media rambat yang digunakannya, yaitu udara. Pantulan gelombang ultrasonik tersebut dapat dimanfaatkan untuk mengukur jarak antara sensor dan benda yang secara ideal dapat dihitung dengan rumus sebagai berikut :

$$s = 0,5. v. t \dots\dots\dots(2.7)$$

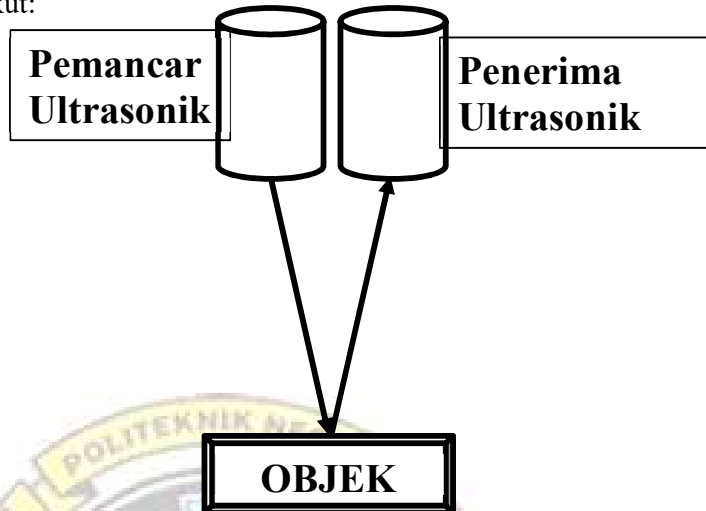
keterangan :

s = jarak objek dengan sensor (m)

v = cepat rambat suara pada medium elastisitas yaitu 343 m/detik

t = waktu tempuh (detik)

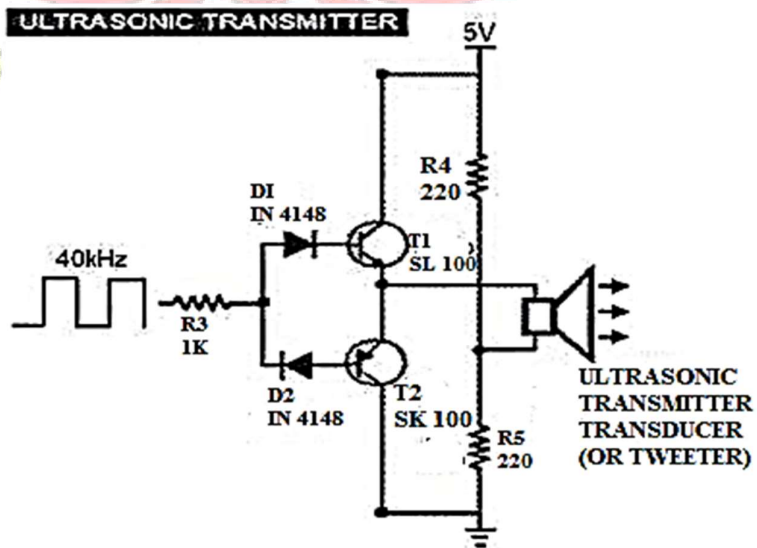
Prinsip pantulan dari sensor ultrasonik ini dapat dilihat pada gambar sebagai berikut:



Gambar 2.12 Prinsip Pemantulan Sensor Ultrasonik

2.5.3 Prinsip Kerja Pemancar Ultrasonik (Transmitter)

Pemancar Ultrasonik ini berupa rangkaian yang memancarkan sinyal sinusoidal berfrekuensi di atas 20 KHz menggunakan sebuah *transducer transmitter* ultrasonik.



Gambar 2.13 Pemancar Ultrasonic Transmitter.

- 1) Sinyal 40 kHz dibangkitkan melalui mikrokontroler.
 - 2) Sinyal tersebut dilewatkan pada sebuah resistor sebesar 3 K ohm untuk pengaman ketika sinyal tersebut membias maju rangkaian dioda dan transistor.
 - 3) Kemudian sinyal tersebut dimasukkan ke rangkaian penguat arus yang merupakan kombinasi dari 2 buah dioda dan 2 buah transistor.
 - 4) Ketika sinyal dari masukan berlogika tinggi (+5V) maka arus akan melewati dioda D_1 (D_1 on), kemudian arus tersebut akan membias transistor T_1 , sehingga arus yang akan mengalir pada kolektor T_1 akan besar sesuai dari penguatan dari transistor.
 - 5) Ketika sinyal dari masukan berlogika tinggi (0 V) maka arus akan melewati dioda D_2 (D_2 ON), kemudian arus tersebut akan membias transistor T_2 , sehingga arus yang akan mengalir pada kolektor T_2 akan besar sesuai dari penguatan dari transistor.
- Resistor R4 dan R6 berfungsi untuk membagi tegangan menjadi 2,5 V. Sehingga pemancar ultrasonik akan menerima tegangan bolak – balik dengan V adalah 5V (+2,5 V s.d -2,5 V).

2.5.4 Prinsip Kerja Penerima Ultrasonik (Receiver)

Penerima Ultrasonik ini akan menerima sinyal ultrasonik yang dipancarkan oleh pemancar ultrasonik dengan karakteristik frekuensi yang sesuai. Sinyal yang diterima tersebut akan melalui proses filterisasi frekuensi dengan menggunakan rangkaian *band pass filter* (penyaring pelewat pita), dengan nilai frekuensi yang dilewatkan telah

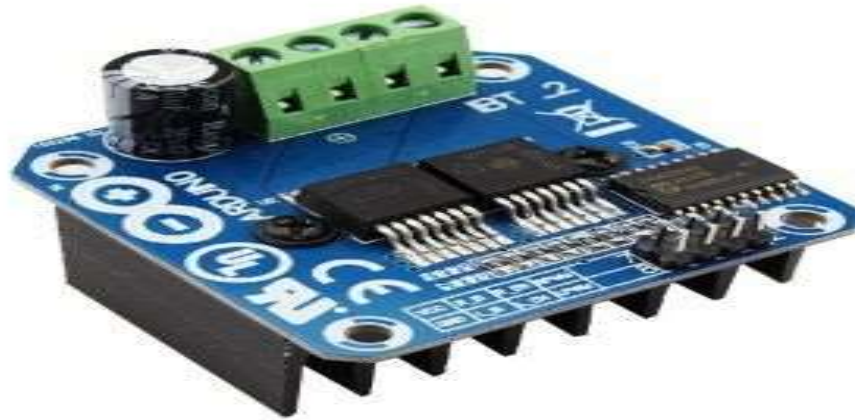
ditentukan. Kemudian sinyal keluarannya akan dikuatkan dan dilewatkan ke rangkaian komparator (pembanding) dengan tegangan referensi ditentukan berdasarkan tegangan keluaran penguat pada saat jarak antara sensor kendaraan mini dengan sekat/dinding pembatas mencapai jarak minimum untuk berbelok arah. Dapat dianggap keluaran komparator pada kondisi ini adalah *high* (logika 1) sedangkan jarak yang lebih jauh adalah *low* (logika 0). Logika-logika biner ini kemudian diteruskan ke rangkaian pengendali (mikrokontroler)



Gambar 2.14 Sensor Ultrasonic HC-SR04

2.6 Driver Motor

Driver motor merupakan suatu rangkaian khusus yang memiliki fungsi untuk mengatur arah dan putaran kecepatan pada motor DC. *Driver* motor DC ini menerapkan sistem *Pulse Width Modulation* (PWM) untuk mengatur kecepatan motor DC. Selain itu *driver* motor ini mampu mengontrol motor DC dengan arus maksimal 43A.



Gambar 2.15 Driver Motor BTS7960

2.7 Bahasa Pemograman MATLAB

Ada beberapa hal dasar yang perlu diketahui dalam Bahasa Pemograman MATLAB yaitu sebagai berikut

2.7.1 Definisi pemograman MATLAB

Matlab merupakan bahasa pemrograman yang hadir dengan fungsi dan karakteristik yang berbeda dengan bahasa pemrograman lain yang sudah ada. Matlab merupakan bahasa pemrograman level tinggi yang dikhususkan untuk kebutuhan komputasi teknis, visualisasi dan pemrograman seperti komputasi matematik, analisis data, pengembangan algoritma, simulasi dan pemodelan dan grafik-grafik perhitungan. (Andrew knight,1999)

2.7.2 Matriks

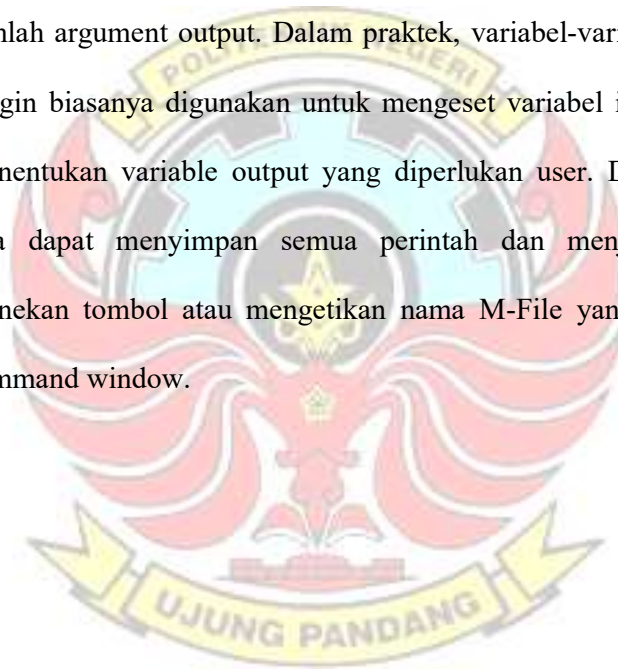
Dapat diasumsikan bahwa didalam matlab setiap data akan disimpan dalam bentuk matriks. Dalam membuat suatu data matriks pada matlab, setiap isi data harus dimulai dari kurung siku '[' dan

diakhiri dengan kurung siku tutup ']'. Untuk membuat variabel dengan data yang terdiri beberapa baris, gunakan tanda 'titik koma' (;) untuk memisahkan data tiap barisnya. Matlab menyediakan beberapa fungsi yang dapat kita gunakan untuk menghasilkan bentuk-bentuk matriks yang diinginkan. Fungsi-fungsi tersebut antara lain untuk membuat matriks yang semua datanya bernilai 0 (Zeros), matriks yang semua datanya bernilai 1 (Ones), matriks dengan data random dengan menggunakan distribusi uniform (Rand), matriks dengan data random dengan menggunakan distribusi normal (Randn), untuk menghasilkan matriks identitas (Eye). (Gunaidi Abdia Away, 2006)

2.7.3 M File

Di dalam matlab, kita dapat menyimpan semua script yang akan digunakan dalam file pada matlab dengan ekstensi M-File dapat dipanggil dengan memilih menu file->new->M-File. Fungsi M-file harus mengikuti beberapa aturan dan sejumlah sifat penting. Aturan-aturan dan sifat-sifat tersebut meliputi nama fungsi dan nama file harus identik misalnya anda membuat fungsi dengan nama pangkat maka anda memberi nama M-file anda pangkat juga. Baris komentar sampai dengan baris bukan komentar yang pertama adalah teks help yang ditampilkan jika anda meminta help dari fungsi yang anda buat. Setiap fungsi mempunyai ruang kerjanya sendiri yang berbeda dengan ruang kerja MATLAB. Satu-satunya hubungan antara ruang kerja matlab dengan variable-variabel dalam fungsi adalah variabel-variabel input

dan dan output fungsi. Jika suatu fungsi mengubah nilai dalam suatu variabel input, perubahan itu hanya tampak dalam fungsi dan tidak mempengaruhi variable yang ada dalam ruang kerja MATLAB. Variabel yang dibuat oleh suatu fungsi tinggal hanya dalam ruang kerja fungsi. Jumlah dari argumen input dan output yang digunakan jika suatu fungsi dipanggil hanya ada dalam fungsi tersebut. Fungsi ruang kerja memuat jumlah argument input. Fungsi kerja nargout memuat jumlah argument output. Dalam praktek, variabel-variabel nargout dan nargin biasanya digunakan untuk mengeset variabel input standar dan menentukan variable output yang diperlukan user. Di dalam M-File, kita dapat menyimpan semua perintah dan menjalankan dengan menekan tombol atau mengetikan nama M-File yang kita buat pada command window.



BAB III METODE PENELITIAN

3.1 Waktu dan Tempat Pelaksanaan

Adapun lokasi pelaksanaan penelitian ini dilakukan di bengkel mekanik Jurusan Teknik Mesin dan Laboratorium Mekatronika dan Sistem Otomasi Politeknik Negeri Ujung Pandang. Dengan waktu perancangan dimulai dari Maret 2018 sampai dengan Agustus 2018.

3.2 Alat dan Bahan yang Digunakan

Dalam metode perancangan ini, diperlukan sejumlah alat dan bahan untuk merakit alat ini sehingga terbentuk sesuai dengan apa yang diinginkan. Adapun alat dan bahan yang digunakan dalam perancangan seperti pada tabel 3.1.

Tabel 3.1 Alat dan Bahan Perancangan

Alat	Bahan
Komputer/Laptop	Sensor Ultrasonik
Solder	Arduino Uno
Penghisap Timah	Pompa Air
Mata Bor	Regulator 5V
Meteran	Power Supply
Bor Tangan	Driver Motor
Cutter	Papan PCB
Elektroda	Acrilic
	Kabel
	Pipa
	Larutan HCl & H ₂ O

3.3 Metode Penelitian/Perancangan

Untuk memperoleh suatu alat yang baik dari segi mutu serta mempertimbangkan segi ekonomis, maka dibutuhkan langkah – langkah seperti dibawah ini.

3.3.1 Studi literatur

Dalam perancangan alat ini, langkah awal yang dilakukan adalah mencari sebanyak – banyaknya data serta informasi melalui berbagai media cetak maupun elektronik, dimana informasi tersebut harus relevan dengan alat yang akan dibuat.

3.3.2 Perancangan *Hardware* dan *Software*

Perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*) meliputi beberapa tahap. Untuk memudahkan dalam perancangan alat ini.

Pembuatan perangkat keras (*hardware*) yaitu pembuatan perangkat *interface* sebagai tempat memasukkan nilai tegangan awal dan tegangan step untuk menjalankan motor yang putarannya diteruskan ke pompa DC.



Gambar 3.1 Perangkat keras *Interface*

Komponen dan spesifikasinya dalam pembuatan perangkat *interface* seperti pada table 3.2.

Tabel 3.2 Komponen dan Spesifikasi

Komponen	Spesifikasi
Mikrokontroller	Arduino Nano
Driver Motor	Resistor 100 Ω , IRFZ244N, PC 117(<i>optocoupler</i>)
LCD	RVI 500K
Push Button	<i>Reset, on-off, dan setting</i>

Sedangkan untuk pembuatan perangkat lunak (*software*) yaitu pembuatan program arduino untuk pembacaan sensor dan pembuatan program matlab untuk menampilkan level ketinggian air dan tegangan motor berupa grafik secara *real time* pada layar komputer.

Adapun listing program Arduino yang berfungsi untuk menampilkan tegangan input, tegangan step dan level pada perangkat interface terdapat pada lampiran 1.

3.3.3 Pengujian dan pengambilan data

Perangkat *interface* yang telah dipasang kemudian dihubungkan dengan komputer melalui kabel USB dan kabel *power supply* dari sumber listrik 220V yang dikonversi menjadi 12 Volt melalui adaptor. Setelah semuanya terhubung, program arduino mengirim program ke perangkat *interface*. Pada proses ini alat *water level control* akan dilakukan pengujian untuk melihat respon dari motor DC ketika

memompa air dari tangki A ke tangki B.

Adapun proses selanjutnya dengan memasukkan tegangan input dan tegangan step pada perangkat *interface*, variabel tegangan output yang keluar dari *driver motor* diukur menggunakan multimeter.

Data yang didapatkan dari tegangan input dan output akan dicatat dalam bentuk tabel kemudian ditampilkan dengan grafik perbandingan antara tegangan *input* dan *output*. Proses pengambilan data ini bertujuan untuk mencari kurva yang tegak lurus untuk dijadikan sebagai nilai operasi.

Perbandingan data tegangan *input* dan *output* dimulai dari tegangan 0-16 Volt dengan setiap kenaikan 0.5 Volt untuk mendapatkan nilai yang sesuai dengan kurva yang padat sebanyak 33 titik berdasarkan pada tabel 4.1

3.3.4 Pengolahan Data

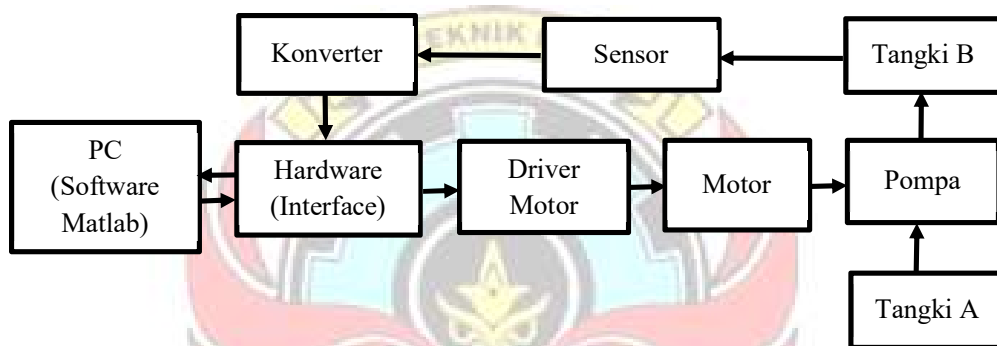
Setelah proses pengambilan data yang dilakukan sebanyak 5 kali dengan tujuan untuk mendapatkan data yang sesuai dengan kurva yang lurus, selanjutnya dilakukan pengolahan data. Variable data tegangan *input* dan *output* akan diolah untuk mencari pemodelan steady state dan step responnya. Selanjutnya akan didapatkan nilai K_{DC} dan persamaan transfer fungsi $H_P(s)$.

Selanjutnya data tersebut akan dimodelkan untuk mencari nilai K_P , K_I , dan K_D dengan metode *trial and error* pada simulink matlab.

3.4 Prosedur Perancangan

3.4.1 Diagram blok

Tahap awal pemodelan eksperimen dan pengontrolan ketinggian air pada tangki ini diperlukan gambaran awal tentang bagaimana sistem kerja dari alat tersebut. Secara garis besar, perancangan menggunakan Sensor Ultrasonik, Komputer, Arduino Uno, Driver Motor, Pompa Air DC, dan Tangki Penampungan.



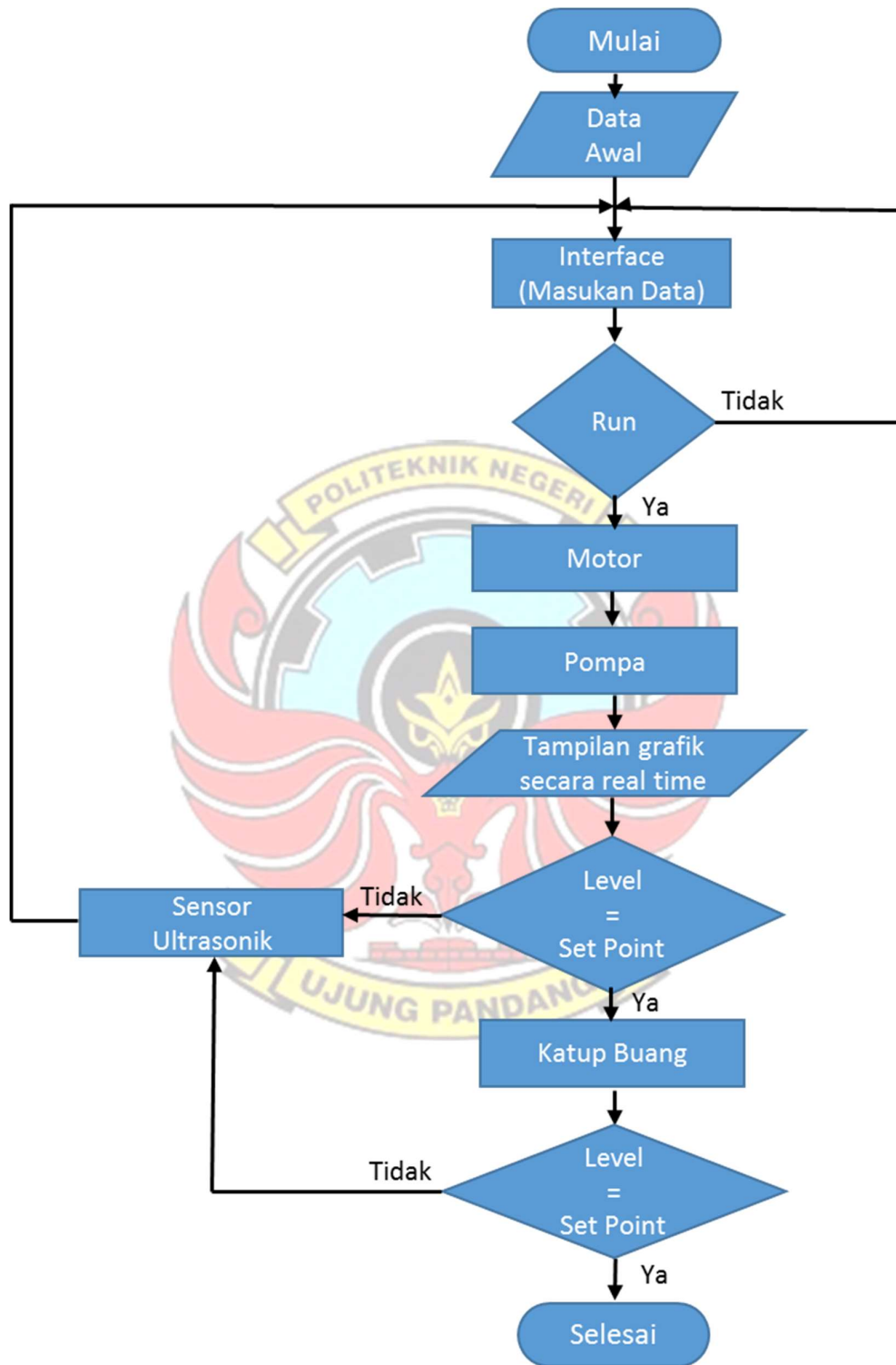
Gambar 3.2 Diagram Blok Sistem Pemodelan Eksperimen dan Pengontrolan Ketinggian Air Pada Tangki

3.4.2 Penentuan Parameter Model Sistem Melalui Pengujian

Menentukan parameter model dari hasil rancangan dengan melakukan pengujian pada alat tersebut.

3.4.3 Diagram Alir

Pembuatan diagram alir ini dibuat untuk mempermudah perancangan program dan untuk membantu melacak kebenaran logika sebuah program. Adapun rancangan diagram alir sistem pemodelan eksperimen dan pengontrolan ketinggian air pada tangki dapat dilihat pada gambar 3.3.

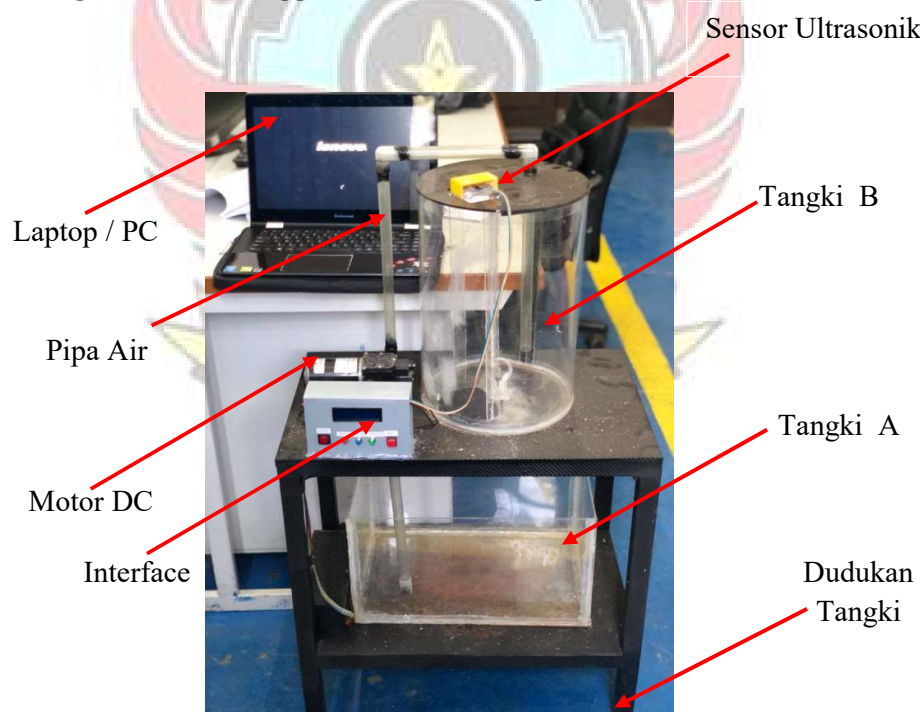


Gambar 3.3 Diagram Alir Sistem Pemodelan Eksperimen dan Pengontrolan Ketinggian Air Pada Tangki

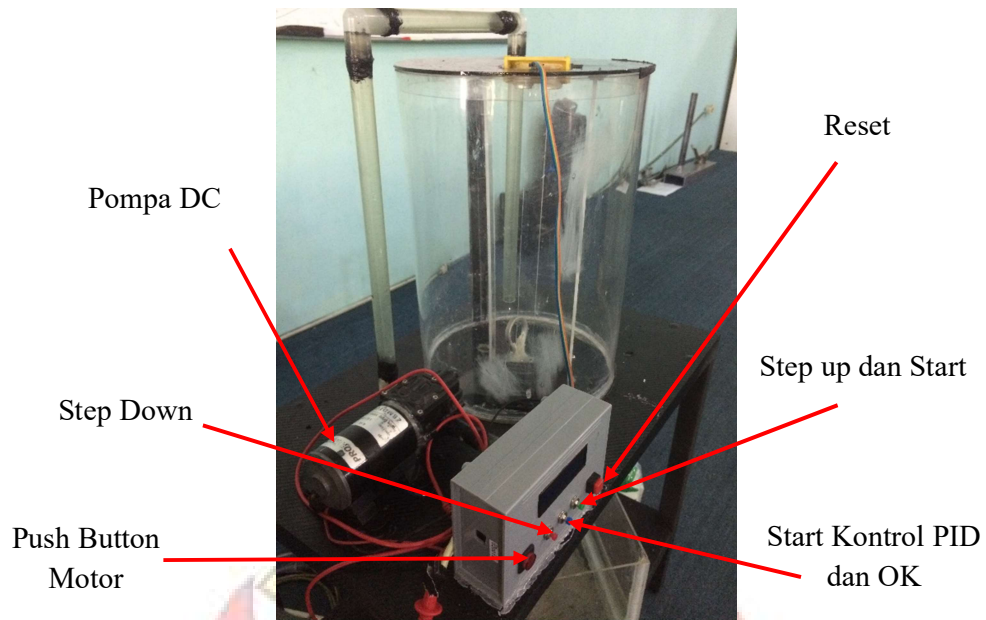
3.5 Prosedur Penelitian

- 1) Menyiapkan peralatan uji dan alat ukur yang akan digunakan.
- 2) Menghubungkan peralatan dengan sumber 220 V AC.
- 3) Menghubungkan mikrokontroler ke komputer dengan menggunakan kabel USB.
- 4) Membuka aplikasi matlab GUI untuk level tangki yang sudah diprogram pada komputer.
- 5) Menyalakan seluruh rangkaian.
- 6) Mengamati setiap kenaikan level air pada tangki A.

3.6 Desain *Water Level Control* pada pemodelan Eksperimen dan Pengontrolan Ketinggian Air Pada Tangki



Gambar 3.4 Alat Sistem Pemodelan Eksperimen dan Pengontrolan Ketinggian Air Pada Tangki



Gambar 3.5 Alat Kontrol Ketinggian Air dengan Perangkat *Interface*



BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil Perancangan

4.1.1 Hasil dan Perancangan Perangkat Keras

Berdasarkan hasil perancangan perangkat *interface* pada alat kontrol ketinggian air, maka alat yang dirancang mampu mengontrol level ketinggian air pada tangki penampungan sesuai level *set point* yang diinginkan. Tegangan masuk pada *interface* akan dikelola sehingga tegangan yang keluar dikirim ke motor untuk menggerakkan pompa, maka air pada tangki A dialirkan ke tangki B, kemudian sensor ultrasonik akan mengirim sinyal ke perangkat *interface* dan ditampilkan secara *real time* dalam bentuk grafik melalui matlab.

Hasil dari perancangan perangkat keras alat ini berupa 1 buah kontrol box berbentuk persegi panjang dengan ukuran panjang 15 cm, lebar 5 cm dan tinggi 10 cm. Pada bagian depan terdapat sebuah layar LCD dan 5 buah tombol yang berfungsi sebagai *on-off*, *start*, *reset* dan *set voltage (input-step)*. Pada bagian dalam terdapat sebuah controller dan *driver motor*. Perangkat *interface* ini juga memiliki fasilitas *power supply* dengan tegangan 12 volt dan 5 volt.

Adapun alat ini juga memiliki beberapa komponen, seperti:

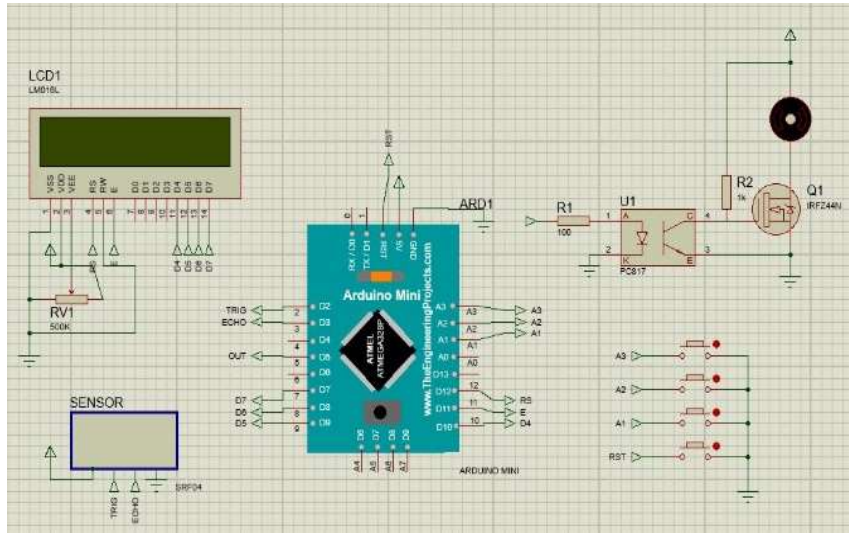
- 1) Arduino Uno, sebagai pemroses data dan pengatur dari seluruh kegiatan sistem yang dibuat.
- 2) Sensor Ultrasonik, sebagai pendeteksi ketinggian air yang akan ditampilkan pada komputer.

- 3) Pompa air DC, berfungsi untuk mengisi air pada tangki penampungan air.
- 4) Rangkain catu daya, berfungsi untuk menurunkan tegangan dari 12 VDC menjadi 5 VDC
- 5) Rangkaian driver motor IRFZ44, berfungsi untuk mengatur arah putaran motor dan kecepatan motor.

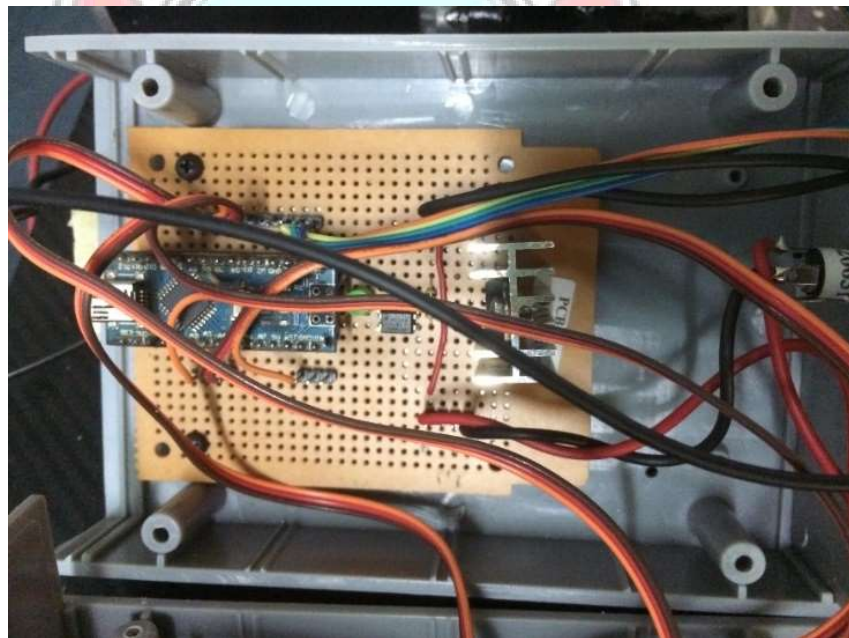
Hasil perancangan dapat dilihat pada gambar berikut:



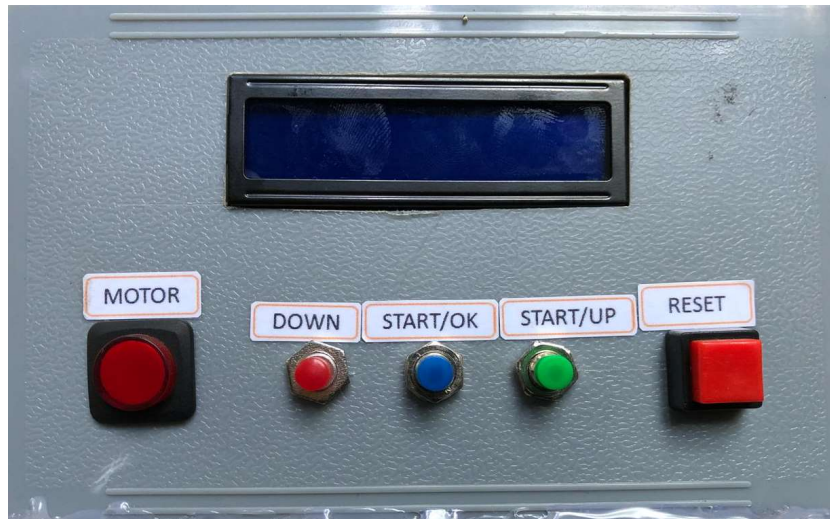
Gambar 4.1 Mekanik Kontrol Level Air



Gambar 4.2 Layout rangkaian Kontrol Level Air



Gambar 4.3 Rangkaian Kontrol Level Air



Gambar 4.4 Perangkat Keras *Interface*

4.1.2 Hasil dan Perancangan Perangkat Lunak

Adapun Listing program Arduino yang telah didesain untuk mengatur nilai tegangan input, tegangan step dan level set point terlihat seperti gambar di bawah ini, Program lebih detail dilampirkan dalam lampiran 1.

Listing program Arduino Mikrokontroller Interface

```
// include the library code:
#include <LiquidCrystal.h>
#include <EEPROM.h>
#include <PID_v1.h>
//Define Variables we'll be connecting to
double Setpoint,Input,Output;
// atur PID disini
double Kp=0, Ki=0, Kd=0;
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 10, 9, 8, 7);
const int analogInPin = A0; // Analog input pin that the
potentiometer is attached to
```

Gambar 4.5 Listing Program Arduino Tegangan Awal dan Step

```

myPID.Compute();
  Input = slevel;
  v_out=map(Output,0,255,255,0);
  teg_out=mapf(Output,0,255,0,12);
  lcd.setCursor(0, 0);
  lcd.print("Lvl");
  lcd.setCursor(6, 0);
  lcd.print("SP");
  lcd.setCursor(11, 0);
  lcd.print("VOUT");
  lcd.setCursor(0, 1);
  lcd.print(slevel);
  lcd.print("%");

```

Gambar 4.6 Listing Program Arduino Pengaturan Nilai PID

4.1.3 Proses Kerja Alat

Tegangan input untuk nilai awal dan nilai step, level set point, nilai K_p , K_i dan K_d dimasukkan pada layar perangkat interface. Program arduino dikirim ke *mikrokontroller* dalam *interface* yang dikirimkan dari komputer melalui kabel USB. Mikrokontroller bekerja dengan mengirimkan sinyal PWM ke driver motor untuk menggerakkan pompa melalui motor, sehingga air pada tangki A akan dialirkan ke tangki B. Selanjutnya level set point juga diproses oleh perangkat interface setelah menerima sinyal dari sensor ultrasonik yang membaca ketinggian air dengan memancarkan gelombang ultrasonik dari *trigger* menuju ke permukaan air, ketika gelombang mengenai permukaan air maka terjadi pantulan gelombang ke *echo* sehingga didapat waktu tempuh antara air dan sensor. Waktu tempuh tersebut akan dikirimkan ke arduino uno lalu diproses sehingga menghasilkan level ketinggian air. Sinyal pulsa dari

sensor ultrasonik akan dikonversi ke tegangan melalui konverter sebagai masukan tambahan ke perangkat interface kemudian dikirim balik ke matlab untuk ditampilkan secara real time level ketinggian air dan tegangan dengan grafik pada matlab.

Sistem pengontrolan pada level ketinggian air menggunakan kontrol *PID* untuk mengatur kerja driver motor, dimana ketika ada penyimpangan level dari set point yang ditentukan maka motor akan *start* dan ketika level set point sudah sesuai maka motor akan *stop*. Adapun penurunan level dapat diatur melalui katup buang dimana air pada tangki B dialirkan ke tangki A.

4.1.4 Cara Pengoperasian Alat

- 1) Hubungkan Power Supply 12VDC dengan sumber 220 VAC
- 2) Koneksikan mikrokontroler ke komputer dengan menggunakan kabel USB
- 3) Buka aplikasi Arduino dan Matlab yang sudah diprogram pada komputer.
- 4) Upload program Arduino tersebut ke mikrokontroler
- 5) Nyalakan seluruh rangkaian dengan menekan tombol start
- 6) Masukkan nilai level set point, tegangan input awal dan tegangan step
- 7) Tekan tombol start untuk menjalankan sistem pengontrolan ketinggian air



Gambar 4.7 Cara Pengoperasian Alat

4.2 Hasil Pengujian dan Pengambilan Data

Pengujian merupakan langkah yang digunakan untuk mengetahui sejauh mana kesesuaian antara rancangan dengan kenyataan pada alat yang telah dibuat, apakah sudah sesuai dengan yang diharapkan atau belum. Pengujian alat ini juga berguna untuk mengetahui tingkat kinerja dari alat tersebut. Setelah dilakukan pengujian, maka hendaknya melakukan pengujian ukuran atau analisa terhadap apa yang diuji untuk mengetahui keberhasilan dari alat yang dibuat dalam tugas akhir ini.

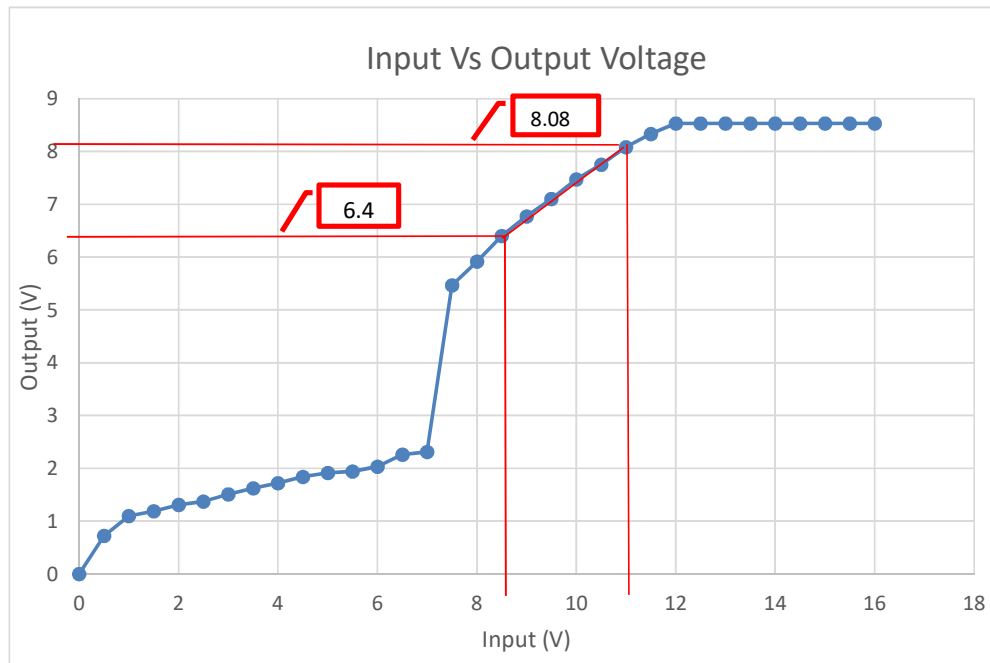
4.2.1 Pemodelan *Steady State*

Menguji alat dari perangkat *interface* dan sistem *water level control* dengan mengambil data untuk menemukan titik operasi yang sesuai dan kestabilan (KDC) dari sistem. Pada tahap ini dilakukan pengukuran terhadap perangkat *interface* untuk melihat hubungan variable nilai input $u(t)$ dan output $y(t)$ dimulai 0 volt sampai 16 V dengan perbedaan 0.5 V setiap datanya. Adapun data yang didapatkan seperti pada table 4.1.

Tabel 4.1 Hubungan Tegangan Input dan Output

No	Tegangan (Volt)		Motor	
	Input	Output	On	Off
1	0	0		✓
2	0.5	0.72		✓
3	1	1.1		✓
4	1.5	1.19		✓
5	2	1.31		✓
6	2.5	1.37		✓
7	3	1.51		✓
8	3.5	1.62		✓
9	4	1.72		✓
10	4.5	1.84		✓
11	5	1.91		✓
12	5.5	1.94		✓
13	6	2.03		✓
14	6.5	2.26		✓
15	7	2.31		✓
16	7.5	5.47	✓	
17	8	5.92	✓	
18	8.5	6.4	✓	
19	9	6.77	✓	
20	9.5	7.1	✓	
21	10	7.47	✓	
22	10.5	7.75	✓	
23	11	8.08	✓	
24	11.5	8.33	✓	
25	12	8.53	✓	
26	12.5	8.53	✓	
27	13	8.53	✓	
28	13.5	8.53	✓	
29	14	8.53	✓	
30	14.5	8.53	✓	
31	15	8.53	✓	
32	15.5	8.53	✓	
33	16	8.53	✓	

Dari tabel data yang telah didapatkan selanjutnya ditampilkan dalam bentuk grafik sebagai berikut.



Gambar 4.8 Hubungan Tegangan Input dan Output

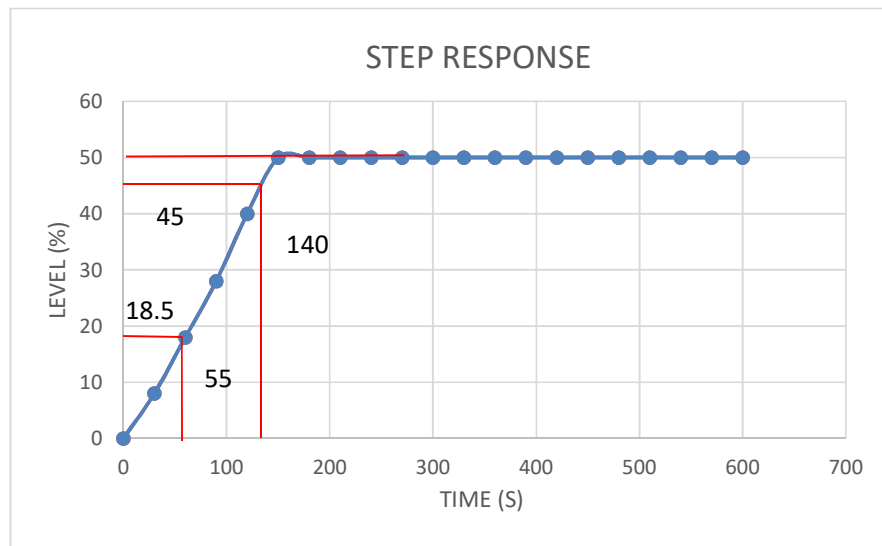
Dari grafik diatas terlihat sebuah garis lurus diantara tegangan input 8.5 dan 11 Volt maka nilai operasi dapat diambil dari nilai tengah yaitu 9.75 volt.

Nilai K_{DC} dapat dihitung dari grafik tersebut

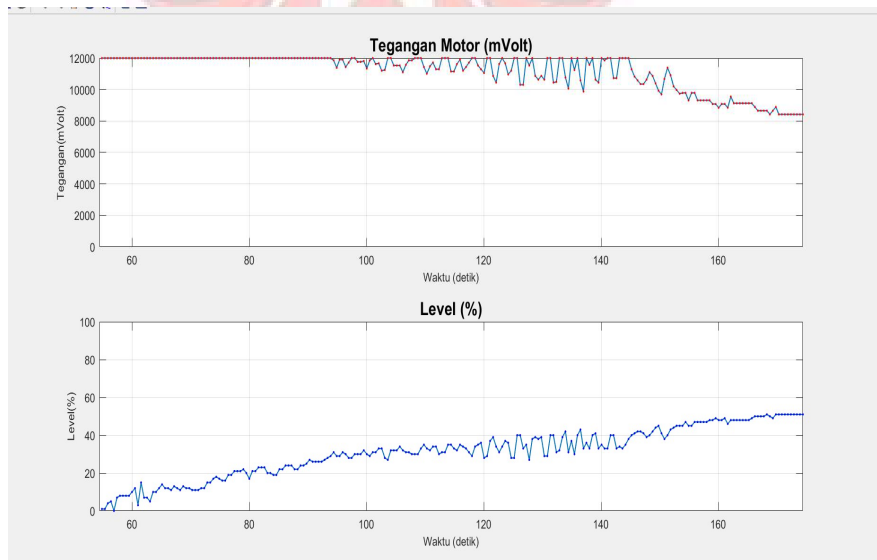
$$K_{DC} = \frac{\Delta output}{\Delta input} = \frac{8.08 - 6.40}{11 - 8.5} = 0.67$$

4.2.2 Step Response

Dari nilai operasi yang telah didapatkan dari grafik yaitu 9.75 Volt, nilai tersebut dijadikan sebagai nilai awal yang akan dimasukkan pada perangkat interface kemudian nilai step 10.75 dengan kenaikan 1 step, maka didapatlah respon sebagai berikut



Gambar 4.9 Hubungan waktu dan level pada tahap step respon



Gambar 4.10 Tampilan trending level dan tegangan pada Matlab

Dari grafik step respon maka didapatkan

$$K_{DC} = 0.67$$

Time constant = 55 detik (63 % from Final Value @ 50.00 %)

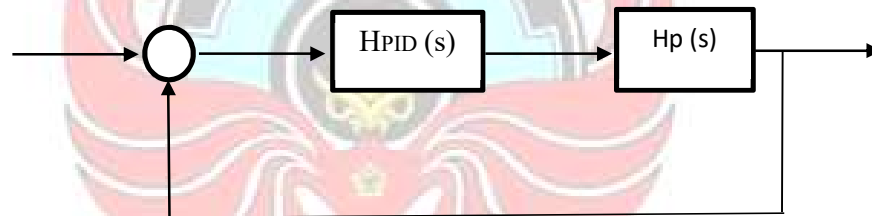
Settling time = 140 detik (10 % from final Value @ 50.00 %)

Dari kedua nilai tersebut didapatkanlah persamaan *transfer function* $H_P(s)$

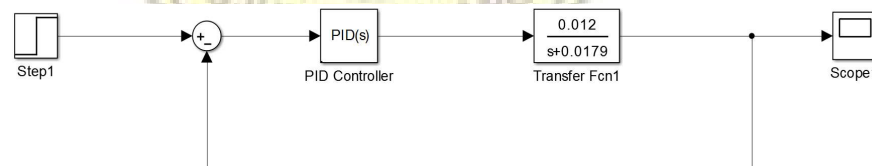
$$H_P(s) = \frac{K_{DC}}{\tau \cdot s + 1} = \frac{0.67}{55s + 1}$$

$$H_P(s) = \frac{0.01218}{s + 0.01818}$$

Dari persamaan transfer function, selanjutnya dimodelkan kemudian diaplikasikan pada Simulink matlab

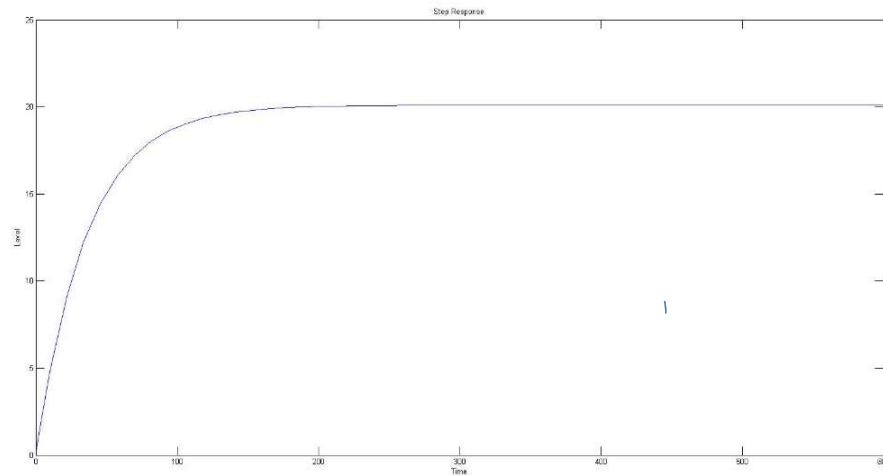


Gambar 4.11 Pemodelan berdasarkan eksperimen



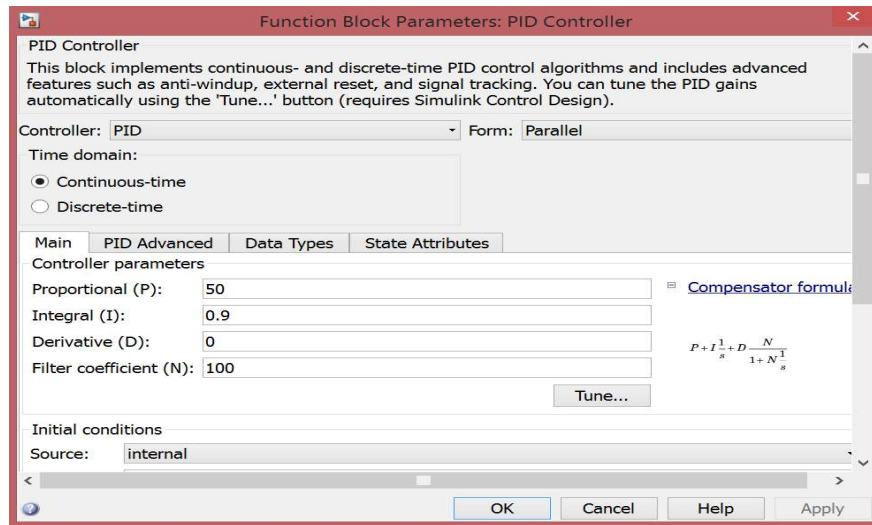
Gambar 4.12 Persamaan transfer fungsi pada Simulink

Dari persamaan transfer fungsi yang yang disumulasikan dalam matlab simulink maka hasil kurva yang didapatkan sesuai dengan gambar 4.13. Hasil ini masih belum dikontrol dengan PID.

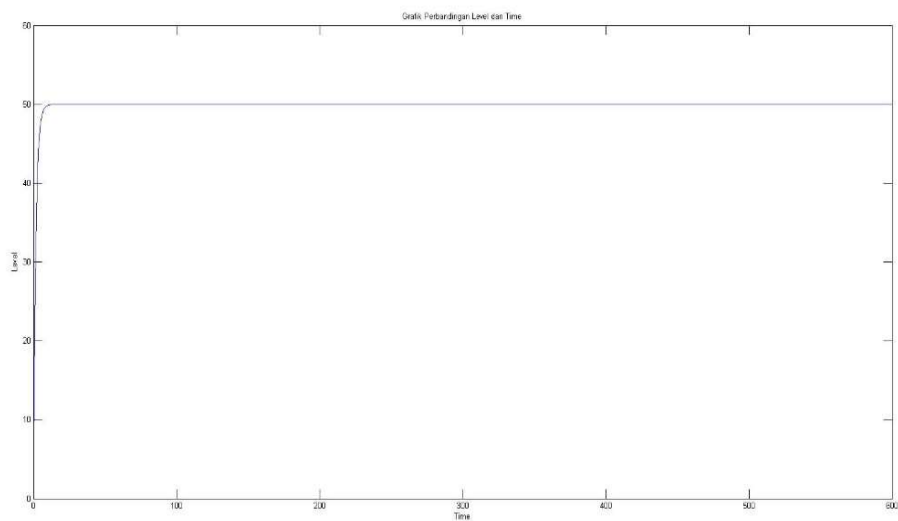


Gambar 4.13 Hasil dari simulasi matlab

Setelah didapatkan rangkaian persamaan transfer fungsi pada simulink yang terdiri dari step, sum, PID, transfer function dan scope. Untuk mendapatkan nilai K_p , K_I , dan K_D berdasarkan dengan trial and error dengan mengatur nilai pada PID Controller. Dari beberapa percobaan, maka didapatkanlah hasil yang sesuai dengan memasukkan nilai *proportional* (P)=50, *Integral* (I)=0.9 dan *Derivative* (D) 0.



Gambar 4.14 Pengaturan Nilai PID pada *Simulink*



Gambar 4.15 Hasil Grafik pada *Simulink*

4.3 Analisa Hasil Pengujian

Tahap pengujian dilakukan sebanyak 2 tahap, yang pertama pengujian terhadap perangkat interface terhadap pengaturan variable nilai tegangan awal dan tegangan step. Pada tahap ini didapatkan data dari keluaran perangkat interface yang diukur dengan multimeter berdasarkan dengan tabel

4.1. Hasil dari data tersebut dapat disimpulkan bahwa pada tegangan 7.5 Volt motor dapat bekerja dan tegangan maksimum yang dapat diukur terbatas sampai 12 Volt sesuai dengan kemampuan motor. Tegangan diatas 12 Volt dapat dikatakan saturasi dengan hasil pengukuran yang sama.

Pengujian tahap kedua terhadap sensor ultrasonik, Grafical User Time (GUI) pada Matlab dan Motor DC. Proses pengambilan data dengan memasukkan nilai tegangan awal 9.75 volt dan tegangan 1 step 10.75 volt berdasarkan gambar 4.1 dimana nilai tersebut diatur pada perangkat *interface*. Selanjutnya memasukkan nilai level set point dengan level 50 % berdasarkan gambar 4.6, dimana *trending* kenaikan level set point dapat dilihat secara real time pada matlab. Saat level mendekati 50 % tegangan keluaran yang dikirim ke motor mulai berkurang sampai dengan motor stop pada level 50%.

Hasil dari kedua pengujian dan pengambilan data tersebut, kemudian diaplikasikan pada Simulink setelah mendapatkan persamaan model yang sesuai agar bisa dikontrol seperti pada gambar 4.11. Sistem water level kontrol ini dapat dikontrol dengan baik dengan mengatur *Proportional*, *Integral* dan *Derivatif* dengan percobaan *trial and error*

Dengan demikian komponen-komponen yang terpasang pada alat dapat difungsikan sesuai dengan fungsinya masing-masing. Sehingga *water level control* dapat bekerja sesuai dengan keinginan.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan perancangan dan pengujian sistem dapat diambil kesimpulan sebagai berikut:

- 1) Perangkat interface terbukti dapat digunakan untuk mengatur nilai tegangan input dan output. Semua sistem bekerja dengan baik dengan terlihatnya tampilan tegangan dan level secara *real time* pada aplikasi matlab di komputer.
- 2) Dari pengujian dan pengambilan data yang dilakukan, dapat diketahui nilai K_{dc} dan persamaan transfer fungsi $H_p(s)$ untuk selanjutnya dimodelkan agar sistem *water level control* dapat dikontrol dengan baik dengan system PID

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan diperoleh beberapa hal yang dapat dijadikan sebagai saran untuk dilakukan pengembangan lebih lanjut yaitu:

- 1) Alat pengontrolan level air ini sebaiknya dapat dikembangkan dengan menggunakan katub solenoid agar keluaran air dapat otomatis.
- 2) Motor DC yang digunakan terbatas pada tegangan 12 Volt, untuk selanjutnya dapat diganti dengan yang lebih besar.
- 3) Agar ditambahkan peredam untuk dudukan motor agar bisa mengurangi getaran.

DAFTAR PUSTAKA

- Arifin, Ilhan. (2015). “*Automatic Water Level Controll Berbasis Mikrokontroller Dengan Sensor Ultrasonik*”. Skripsi. Fakultas Teknik Elektro Universitas Negeri Semarang
- Antoni Susiono, et all. (2006). *Aplikasi Scada System pada Miniatur Water Level Control*. Jurnal Elektro (Nomor 2 volume 6). Hlm.37.
- Budiarso Z, Nurraharzo E. 2011. *Sistem Monitoring Ketinggian Air Bendungan Berbasis Mikrokontroler*. Jurnal Dinamika Informatika.
- Djuandi, Feri (2011). “*Pengenalan Arduino*”. www.Tobuku.com. Diakses pada tanggal 26 januari 2018
- Puluhulawa Irman, Alam Mangku (2016). “*Rancang Bangun Kontrol Level Air Berbasis Mikrokontroller Menggunakan Sensor Ultrasonik*” Skripsi. Jurusan Teknik Mesin Politeknik Negeri Ujung Pandang
- Surindra, M. Deni. “*Sistem Kontrol Level Air Pada Feed Water Tank Menggunakan Mikrokontroler ATmega8535 Dengan Delphi Sebagai Media Tampilannya*”. Jurusan Teknik Mesin Politeknik Negeri Semarang.
- Suastika, K.G, dkk (2013). “*Sensor Ultrasonik Sebagai Alat Pengukur Kecepatan Aliran Udara Dalam Pipa*”. Dalam jurnal Pendidikan Fisika Indonesia. Universitas palangkaraya.
- Gultom, Iwan Primus (2013). <http://iwangoeltom.blogspot.co.id/2013/08/makalah-mat-lab.html>. Diakses pada tanggal 26 januari 2018
- Susanti, Lely https://www.academia.edu/6520482/Makalah_Tugas_Rancang_Water_Level_Control. Diakses pada tanggal 26 januari 2018
- <http://www.elangsakti.com/2015/05/sensor-ultrasonik.html>. Diakses pada tanggal 28 januari 2018

LAMPIRAN 1

LISTING PROGRAM



Listing program Arduino Mikrokontroller Interface

```
// include the library code:

#include <LiquidCrystal.h>

#include <EEPROM.h>

#include <PID_v1.h>

//Define Variables we'll be connecting to
double Setpoint,Input,Output;

// atur PID disini
double Kp=0, Ki=0, Kd=0;

PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 10, 9, 8, 7);

const int analogInPin = A0; // Analog input pin that the potentiometer is
attached to

const int but_r = A1;
const int but_b = A2;
const int but_g = A3;

const int analogOutPin = 5; // Analog output pin that the LED

const int trigPin = 3;
const int echoPin = 2;

// defines variables
```

```

unsigned long duration;

float distance;

unsigned int sensorValue = 0;    // value read from the pot
unsigned char outputValue = 0;   // value output to the PWM (analog out)
float vawal,vsstep;
unsigned char awal, sstep, kkp, kki, kkd;
unsigned char pwmawal, pwmsstep, teg_out, v_out;
unsigned int svawal,svsstep,slevel;
unsigned char jarak=4;

void setup() {
    // initialize serial communications at 9600 bps:
    Serial.begin(9600);
    // set up the LCD's number of columns and rows:
    lcd.begin(16, 2);

    pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
    pinMode(echoPin, INPUT); // Sets the echoPin as an Input

    // initialize the pushbutton pin as an input:
    pinMode(analogInPin, INPUT);digitalWrite(analogInPin, LOW);
    pinMode(but_r, INPUT);digitalWrite(but_r, HIGH);
    pinMode(but_b, INPUT);digitalWrite(but_b, HIGH);
    pinMode(but_g, INPUT);digitalWrite(but_g, HIGH);

```

```

awal = EEPROM.read(0);
sstep = EEPROM.read(1);
Setpoint = EEPROM.read(2)-10;
//kcp = EEPROM.read(3);
//Kp=mapf(kcp,0,255,0,10);
//kki = EEPROM.read(4);
//Ki=mapf(kki,0,255,0,10);
//kcd = EEPROM.read(4);
//Kd=mapf(kcd,0,255,0,10);
//Setpoint = 50;
Input = slevel;
//turn the PID on

myPID.SetMode(AUTOMATIC);
}

void loop() {
  analogWrite(analogOutPin, 255);
  lcd.setCursor(0, 0);
  lcd.print("  Water Level ");
  lcd.setCursor(0, 1);
  lcd.print("  Mulai  ");
  Serial.println("0,0");
  delay(500);

  if(digitalRead(but_r)==LOW)
  {

```



```

analogWrite(analogOutPin, 255);

lcd.setCursor(0, 0);

lcd.print(" Atur Parameter ");

lcd.setCursor(0, 1);

lcd.print(" Awal & SP PID ");

delay(1000);

while(1)
{
  lcd.setCursor(0, 0);
  lcd.print(" Atur Parameter ");
  lcd.setCursor(0, 1);
  lcd.print("Volt OK SP PID");
  delay(200);
  lcd.clear();
  if(digitalRead(but_r)==LOW)
  {
    while(1)
    {
      lcd.setCursor(0, 0);
      lcd.print("Atur Nilai Awal ");
      vawal=mapf(awal,0,255,0,12);
      lcd.setCursor(5, 1);
      lcd.print(vawal);
      if(digitalRead(but_r)==LOW)
      {
        awal--;
      }
    }
  }
}

```

```

if(digitalRead(but_g)==LOW)
{
    awal++;
}
if(digitalRead(but_b)==LOW)
{
    lcd.setCursor(0, 1);
    lcd.print("Menyimpan Nilai ");
    EEPROM.write(0, awal);
    delay(1000);
    break;
}
delay(200);
lcd.clear();
}

////////////////////////////////////
while(1)
{
    lcd.setCursor(0, 0);
    lcd.print("Atur Nilai Step ");
    vsstep=mapf(sstep,0,255,0,12);
    lcd.setCursor(5, 1);
    lcd.print(vsstep);
    if(digitalRead(but_r)==LOW)
    {
        sstep--;
    }
}

```



```

    }
    if(digitalRead(but_g)==LOW)
    {
        sstep++;
    }
    if(digitalRead(but_b)==LOW)
    {
        if(awal>sstep)
        {
            lcd.setCursor(0, 0);
            lcd.print("Nilai Awal<Step ");
            delay(1000);
        }
        else
        {
            lcd.setCursor(0, 1);
            lcd.print("Menyimpan Nilai ");
            EEPROM.write(1, sstep);
            delay(1000);
            break;
        }

    }

    delay(200);
    lcd.clear();
}
////////////////////

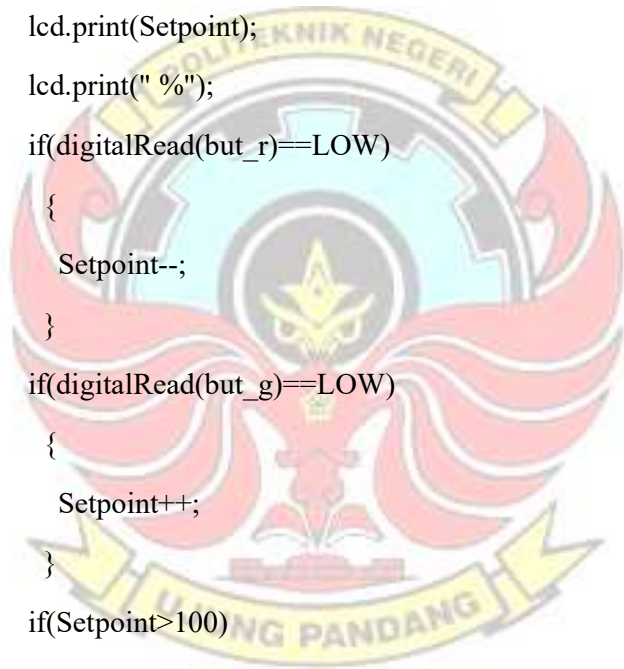
```



```

}
if(digitalRead(but_g)==LOW)
{
while(1)
{
lcd.setCursor(0, 0);
lcd.print(" Atur Set Point ");
lcd.setCursor(4, 1);
lcd.print(Setpoint);
lcd.print(" %");
if(digitalRead(but_r)==LOW)
{
Setpoint--;
}
if(digitalRead(but_g)==LOW)
{
Setpoint++;
}
if(Setpoint>100)
{
Setpoint=100;
}
if(Setpoint<0)
{
Setpoint=0;
}
if(digitalRead(but_b)==LOW)

```



```

        {
            lcd.setCursor(0, 1);
            lcd.print("Menyimpan Nilai ");
            EEPROM.write(2, Setpoint);
            delay(1000);
            break;
        }
        delay(200);
        lcd.clear();
    }
    //////////////////////////////////////
}
if(digitalRead(but_b)==LOW)
{
    lcd.setCursor(0, 0);
    lcd.print(" Menyimpan ");
    lcd.setCursor(0, 1);
    lcd.print(" Semua Nilai ");
    delay(1000);
    break;
}
}

if(digitalRead(but_g)==LOW)
{

```



```

while(1)
{
  if(digitalRead(but_b)==LOW)
  {
    while(1)
    {
      ///////////////////////////////////
      digitalWrite(trigPin, LOW);
      delayMicroseconds(2);
      // Sets the trigPin on HIGH state for 10 micro seconds
      digitalWrite(trigPin, HIGH);
      delayMicroseconds(10);
      digitalWrite(trigPin, LOW);
      // Reads the echoPin, returns the sound wave travel time in
microseconds
      duration = pulseIn(echoPin, HIGH);
      // Calculating the distance
      distance= (duration*0.034/2)-jarak;
      // Prints the distance on the Serial Monitor
      if(distance>27)
      {
        distance=27;
      }
      if(distance<0)
      {
        distance=0;
      }
    }
  }
}

```

```

slevel=mapf(distance,0,30,100,0);
////////////////////////////////////
lcd.setCursor(0, 0);
lcd.print("Level");
lcd.setCursor(6, 1);
lcd.print("VOUT");
lcd.setCursor(11, 0);
lcd.print(vawal);
lcd.setCursor(0, 1);
lcd.print(slevel);
lcd.print("%");
lcd.setCursor(11, 1);
lcd.print(vstep);
pwmstep=map(sstep,0,255,255,0);
analogWrite(analogOutPin, pwmstep);
Serial.print(svsstep);
Serial.print(",");
Serial.println(slevel);
delay(500);
lcd.clear();
}
}
vawal=mapf(awal,0,255,0,12);
svawal=map(awal,0,255,0,12000);
vstep=mapf(sstep,0,255,0,12);
svsstep=map(sstep,0,255,0,12000);
////////////////////////////////////

```

```

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in
microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= (duration*0.034/2)-jarak;
// Prints the distance on the Serial Monitor

if(distance>27)
{
  distance=27;
}
if(distance<0)
{
  distance=0;
}

slevel=mapf(distance,0,27,100,0);
////////////////////

lcd.setCursor(0, 0);
lcd.print("Level");
lcd.setCursor(6, 0);

```

```

    lcd.print("VOUT");
    lcd.setCursor(11, 0);
    lcd.print(vawal);
    lcd.setCursor(0, 1);
    lcd.print(slevel);
    //lcd.print(distance);
    lcd.print("%");
    lcd.setCursor(11, 1);
    lcd.print(vsstp);

    pwmawal=map(awal,0,255,255,0);
    analogWrite(analogOutPin, pwmawal);
    Serial.print(svawal);
    Serial.print(",");
    Serial.println(slevel);
    delay(500);
    lcd.clear();
}
}
if(digitalRead(but_b)==LOW)
{
    while(1)
    {
        digitalWrite(trigPin, LOW);
        delayMicroseconds(2);
        // Sets the trigPin on HIGH state for 10 micro seconds

```



```

digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in
microseconds

duration = pulseIn(echoPin, HIGH);

// Calculating the distance
distance= (duration*0.034/2)-jarak;

// Prints the distance on the Serial Monitor

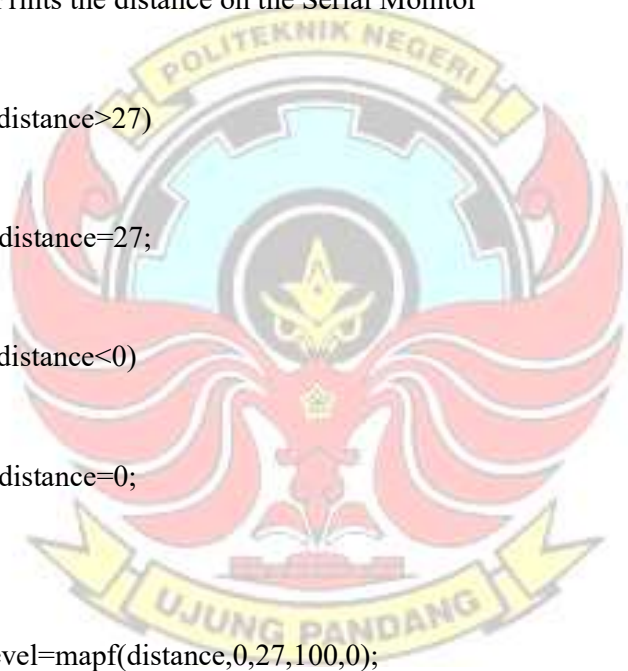
if(distance>27)
{
  distance=27;
}
if(distance<0)
{
  distance=0;
}

slevel=mapf(distance,0,27,100,0);

////////////////////

myPID.Compute();
Input = slevel;
v_out=map(Output,0,255,255,0);
teg_out=mapf(Output,0,255,0,12);
lcd.setCursor(0, 0);

```



```
lcd.print("Lvl");  
lcd.setCursor(6, 0);  
lcd.print("SP");  
lcd.setCursor(11, 0);  
lcd.print("VOUT");  
lcd.setCursor(0, 1);  
lcd.print(slevel);  
lcd.print("%");  
lcd.setCursor(5, 1);  
lcd.print(Setpoint);  
lcd.setCursor(11, 1);  
lcd.print(teg_out);  
analogWrite(analogOutPin, v_out);  
svawal=map(Output,0,255,0,12000);  
  
Serial.print(svawal);  
Serial.print(",");  
Serial.println(slevel);  
  
delay(500);  
lcd.clear();  
}  
}
```



```

/*
sensorValue = analogRead(analogInPin);
lcd.setCursor(0, 0);
lcd.print(sensorValue);
if(digitalRead(but_r)==LOW)
{
    outputValue--;
}
if(digitalRead(but_g)==LOW)
{
    outputValue++;
}
lcd.setCursor(0, 1);
lcd.print(outputValue);
// change the analog out value:
analogWrite(analogOutPin, outputValue);
tegangan=mapf(outputValue,0,255,0,12);
lcd.setCursor(8, 1);
lcd.print(tegangan);
delay(200);
lcd.clear();
// map it to the range of the analog out:
*/
}

double mapf(double val, double in_min, double in_max, double out_min,
double out_max) {

```

LAMPIRAN 2

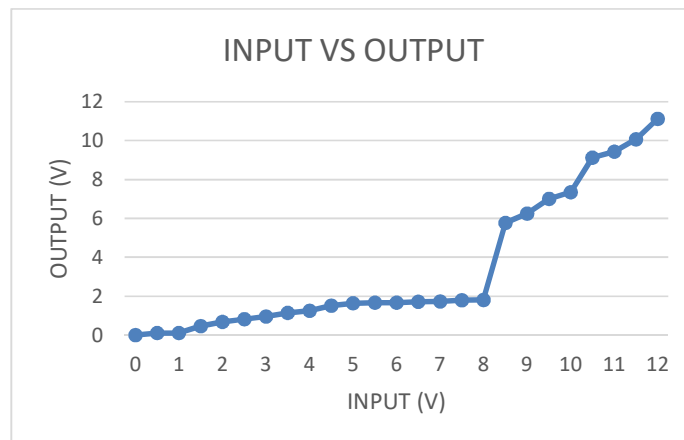
The logo of Politeknik Negeri Ujung Pandang is a circular emblem. It features a central gear with a yellow bird-like figure in the center. The gear is surrounded by a red and white flame-like border. The text "POLITEKNIK NEGERI" is written in yellow on a banner at the top, and "UJUNG PANDANG" is written in yellow on a banner at the bottom.

HASIL PENGUJIAN DAN PENGOLAHAN DATA

1. Hasil Pengukuran Tegangan Input dan Output

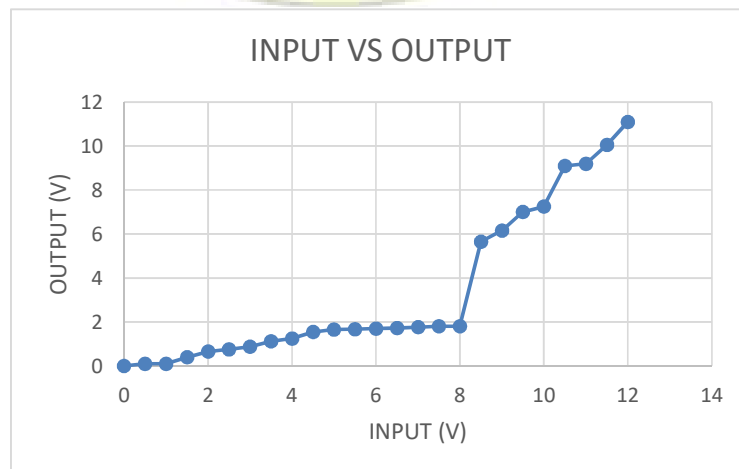
1.1 Pengukuran Percobaan 1

No	Tegangan (Volt)	
	Input	Output
1	0	0
2	0.5	0.11
3	1	0.11
4	1.5	0.47
5	2	0.68
6	2.5	0.82
7	3	0.96
8	3.5	1.15
9	4	1.26
10	4.5	1.51
11	5	1.64
12	5.5	1.67
13	6	1.67
14	6.5	1.72
15	7	1.74
16	7.5	1.8
17	8	1.81
18	8.5	5.76
19	9	6.25
20	9.5	7.01
21	10	7.34
22	10.5	9.12
23	11	9.43
24	11.5	10.07
25	12	11.12



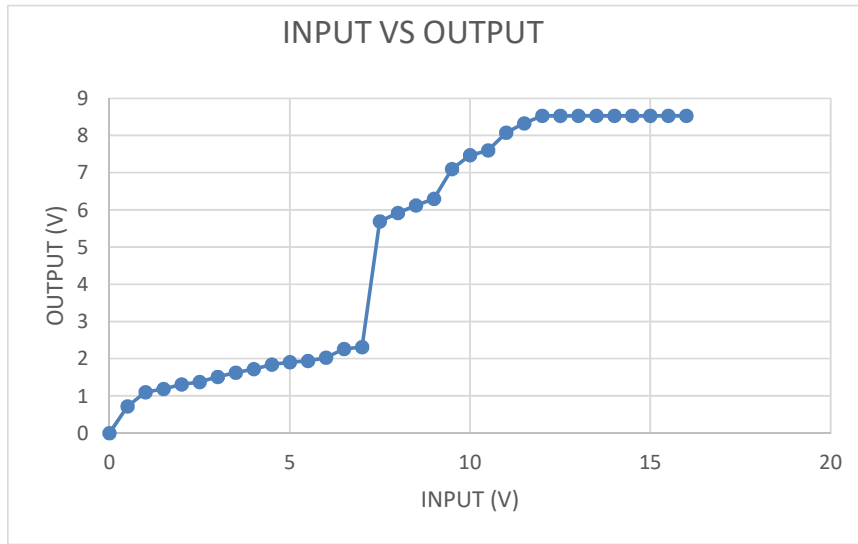
1.2 Pengukuran Percobaan 2

No	Tegangan (Volt)	
	Input	Output
1	0	0
2	0.5	0.09
3	1	0.10
4	1.5	0.39
5	2	0.65
6	2.5	0.75
7	3	0.88
8	3.5	1.12
9	4	1.25
10	4.5	1.55
11	5	1.65
12	5.5	1.67
13	6	1.69
14	6.5	1.72
15	7	1.76
16	7.5	1.80
17	8	1.81
18	8.5	5.65
19	9	6.15
20	9.5	7.00
21	10	7.25
22	10.5	9.10
23	11	9.19
24	11.5	10.05
25	12	11.10



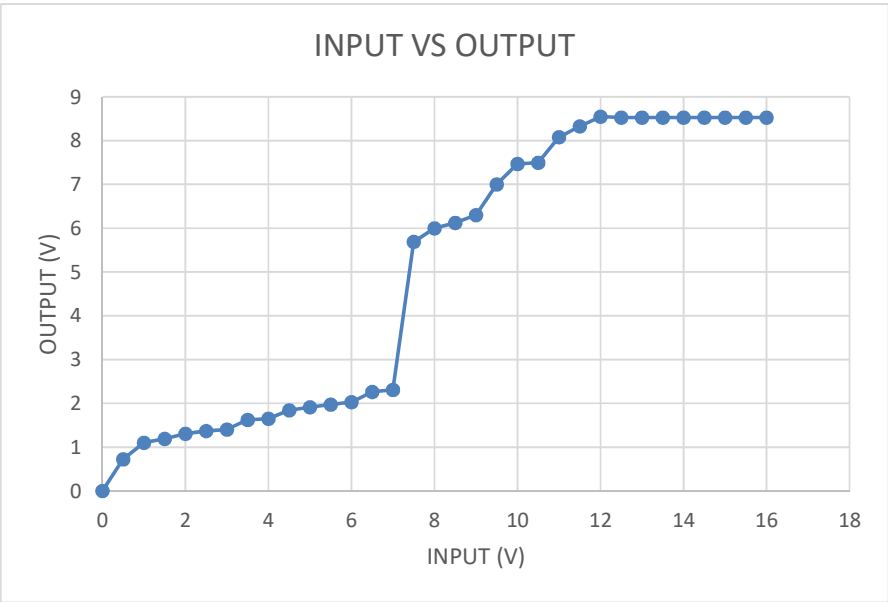
1.3 Pengukuran Percobaan 3

No	Tegangan (Volt)	
	Input	Output
1	0	0
2	0.5	0.72
3	1	1.1
4	1.5	1.19
5	2	1.31
6	2.5	1.37
7	3	1.51
8	3.5	1.62
9	4	1.72
10	4.5	1.84
11	5	1.91
12	5.5	1.94
13	6	2.03
14	6.5	2.26
15	7	2.31
16	7.5	5.69
17	8	5.92
18	8.5	6.12
19	9	6.3
20	9.5	7.1
21	10	7.47
22	10.5	7.6
23	11	8.08
24	11.5	8.33
25	12	8.53
26	12.5	8.53
27	13	8.53
28	13.5	8.53
29	14	8.53
30	14.5	8.53
31	15	8.53
32	15.5	8.53
33	16	8.53



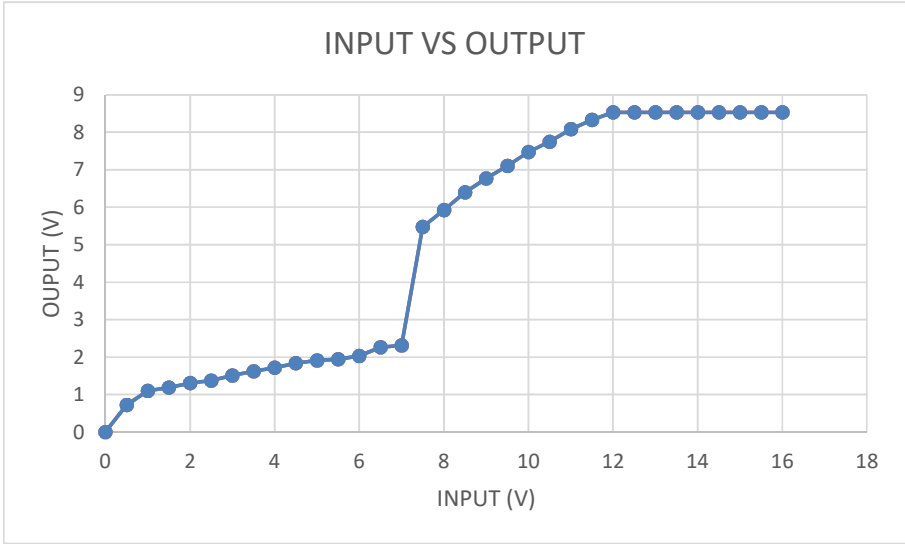
1.4 Pengukuran Percobaan 4

No	Tegangan (Volt)	
	Input	Output
1	0	0
2	0.5	0.72
3	1	1.1
4	1.5	1.19
5	2	1.31
6	2.5	1.37
7	3	1.4
8	3.5	1.62
9	4	1.65
10	4.5	1.84
11	5	1.91
12	5.5	1.97
13	6	2.03
14	6.5	2.26
15	7	2.31
16	7.5	5.69
17	8	6
18	8.5	6.12
19	9	6.3
20	9.5	7
21	10	7.47
22	10.5	7.5
23	11	8.08
24	11.5	8.33
25	12	8.53
26	12.5	8.53
27	13	8.53
28	13.5	8.53
29	14	8.53
30	14.5	8.53
31	15	8.53
32	15.5	8.53
33	16	8.53



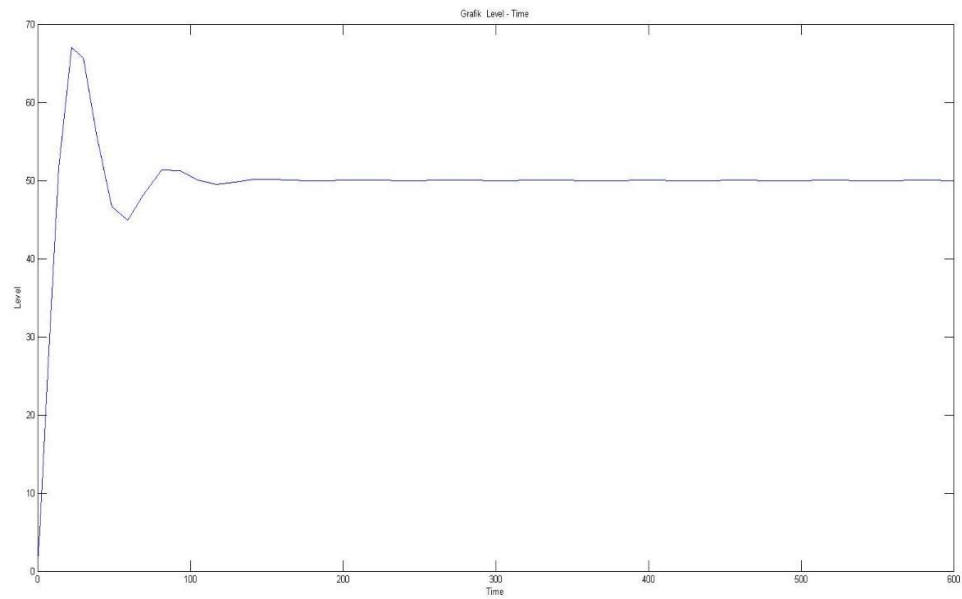
1.5 Pengukuran Percobaan 5

No	Tegangan (Volt)	
	Input	Output
1	0	0
2	0.5	0.72
3	1	1.1
4	1.5	1.19
5	2	1.31
6	2.5	1.37
7	3	1.51
8	3.5	1.62
9	4	1.72
10	4.5	1.84
11	5	1.91
12	5.5	1.94
13	6	2.03
14	6.5	2.26
15	7	2.31
16	7.5	5.47
17	8	5.92
18	8.5	6.4
19	9	6.77
20	9.5	7.1
21	10	7.47
22	10.5	7.75
23	11	8.08
24	11.5	8.33
25	12	8.53
26	12.5	8.53
27	13	8.53
28	13.5	8.53
29	14	8.53
30	14.5	8.53
31	15	8.53
32	15.5	8.53
33	16	8.53

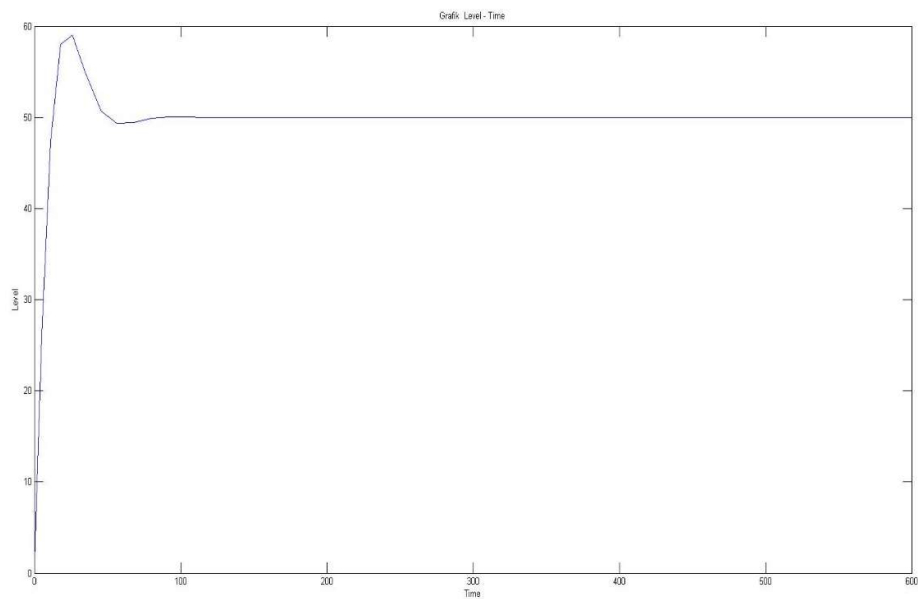


2. Hasil Pengaturan Nilai PID Pada Simulink

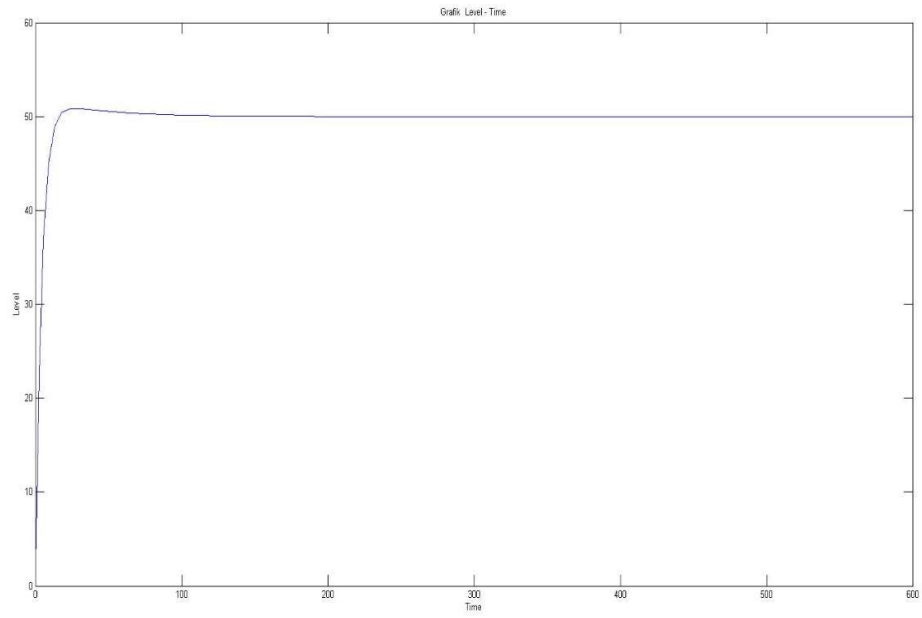
2.1.Percobaan 1 : Kp 5, Ki 1 dan Kd 0



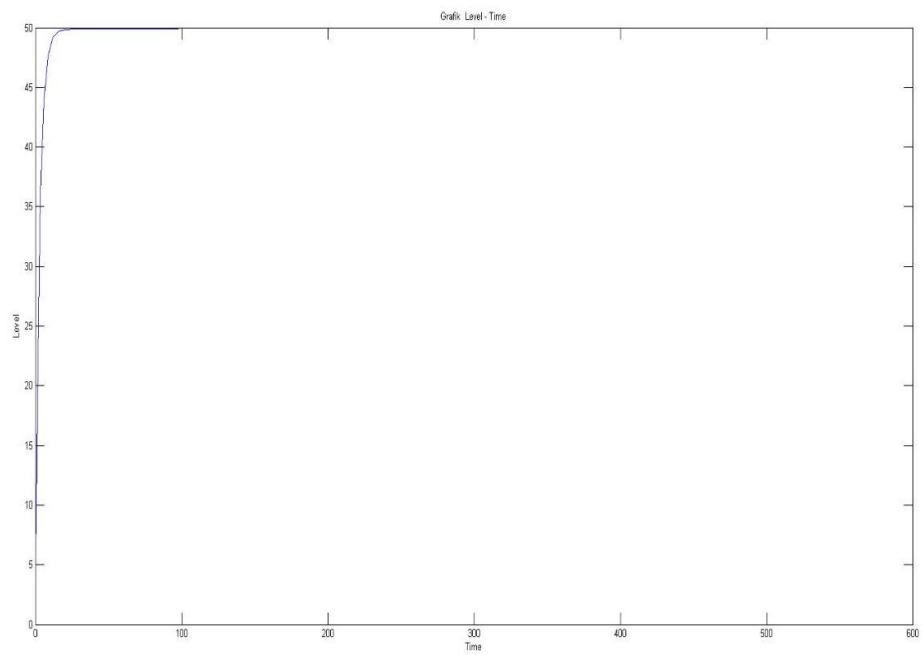
2.2.Percobaan 2 : Kp 10, Ki 1 dan Kd 0



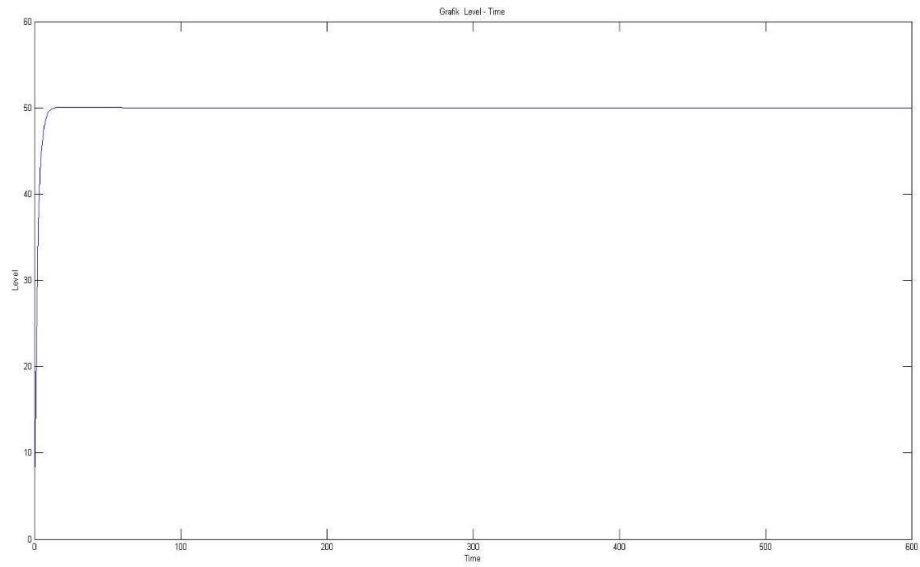
2.3.Percobaan 3 : Kp 20, Ki 0.5 dan Kd 0



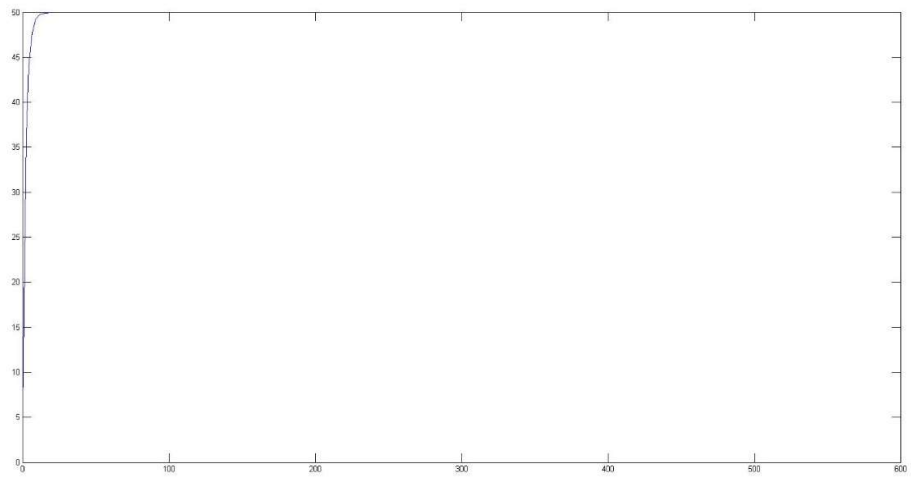
2.4.Percobaan 4 : Kp 30 Ki 0.5 dan Kd 0



2.5. Percobaan 5 : Kp 40, Ki 0.75 dan Kd 0



2.6. Percobaan 5 : Kp 50, Ki 0.9 dan Kd

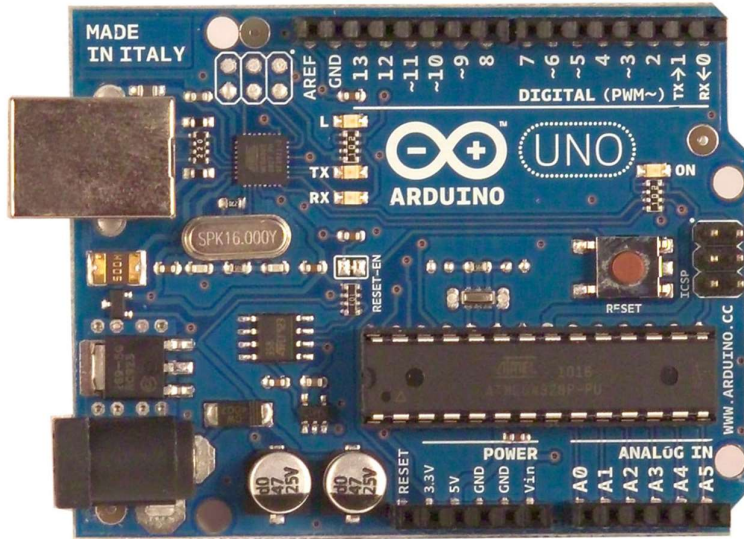


LAMPIRAN 3

The logo of Politeknik Negeri Ujung Pandang is a circular emblem. It features a central gear with a yellow bird-like figure in the center. The gear is surrounded by a red and white flame-like border. The text "POLITEKNIK NEGERI" is written in yellow on a banner at the top, and "UJUNG PANDANG" is written in yellow on a banner at the bottom.

**DATASHEET ARDUINO
DAN
SENSOR ULTRASONIK**

Arduino UNO



ΠΛΟΗΚΤΑ ΟΛΕΛΛΙΣΜ

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, 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 Uno 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.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Index

Technical Specifications

Page 2

How to use Arduino
Programming Environment, Basic Tutorials

Page 6

Terms & Conditions

Page 7



radiospares

RADIONICS



Technical Specification

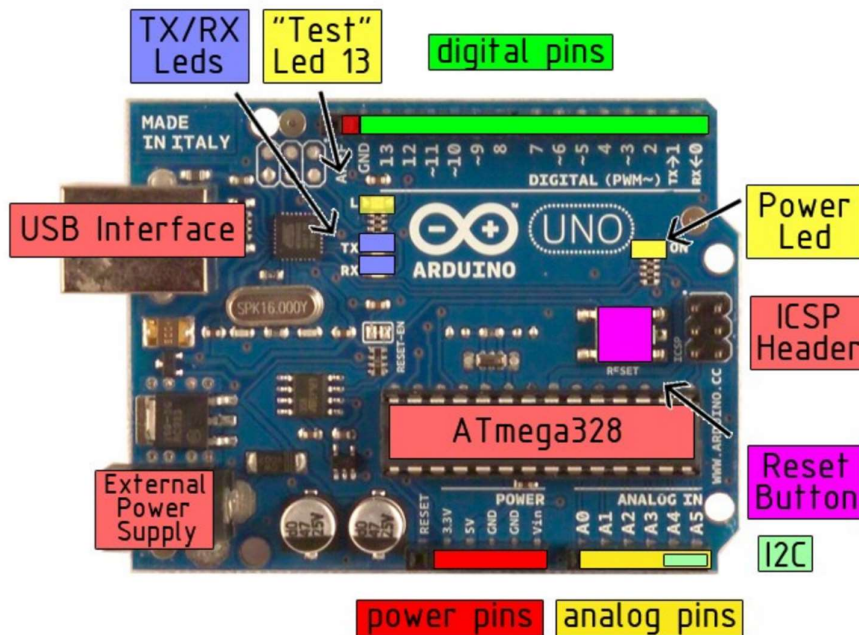


EAGLE files: [arduino-duemilanove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

Summary

Microcontroller	ATmega328
Operating Voltage	
5V Input Voltage (recommended)	
7-12V Input Voltage (limits)	
6-20V	
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	
2 KB EEPROM	
1 KB Clock Speed	
16 MHz	





Radiospares

RADIONICS



The Arduino Uno 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 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.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno 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). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 and 3. 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: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

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.



radiospares RADIONICS



The Uno has 6 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 the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

I²C: 4 (SDA) and 5 (SCL). Support I²C (TWI)

communication using the [Wire library](#). 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.

See also the [mapping between Arduino pins and Atmega328 ports](#).

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial 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 Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno 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 . 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).



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno 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 ATmega328P 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 Uno 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 Uno. 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 Uno 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.

USB Overcurrent Protection

The Arduino Uno 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.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 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.

How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install Windows Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

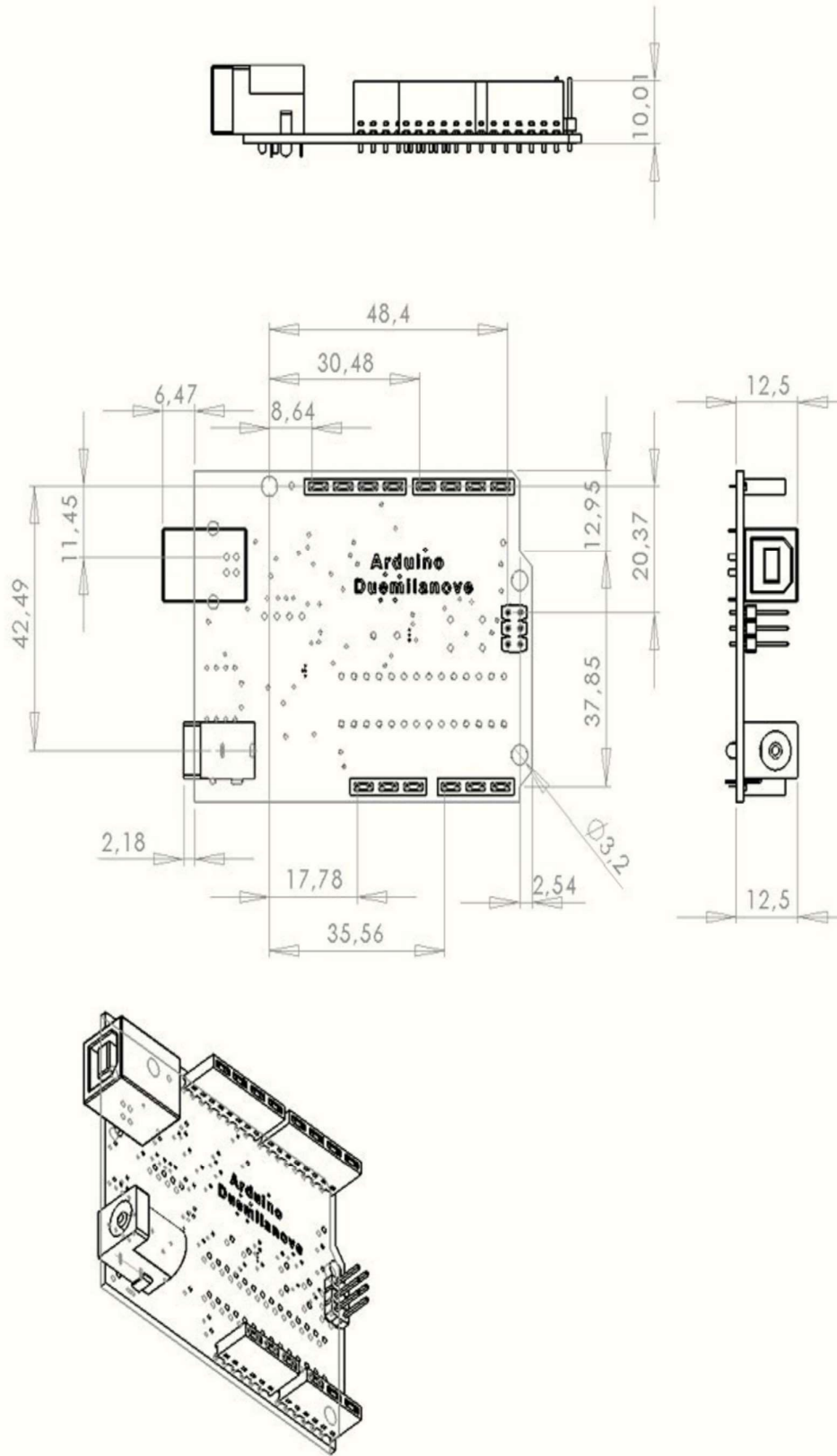
Press Compile button (to check for errors)

Upload

TX RX Flashing

Blinking Led!

Dimensioned Drawing



Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



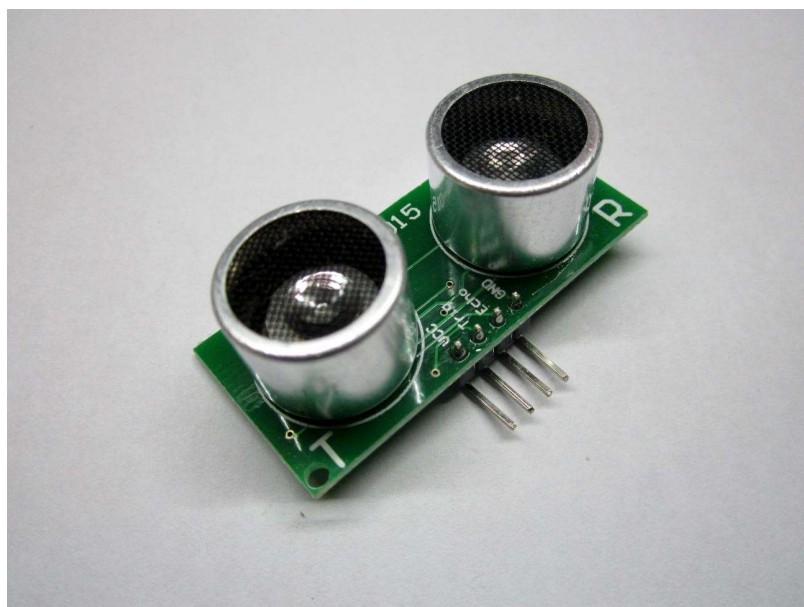
Environmental Policies



The producer of Arduino has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



US-015 Ultrasonic Distance Measuring Sensor Module



US-015 ultrasonic module can realize 2 ~ 700cm of non-contact ranging function, has 5 V of wide voltage input range, static power consumption is less than 3mA. Their own temperature sensor will revise the ranging and has many communication modes like GPIO, just like watchdog inside, very stable and reliable.

LAMPIRAN 4



**DOKUMENTASI
PENGUJIAN ALAT
DAN
PENGAMBILA DATA**

1. Pengujian Alat dan pengambilan Data Tegangan Input dan Output



2. Memasukkan Data pada Perangkat Interface



3. Mengamati Performa Motor dan Tampilan Kenaikan Level

