

BAB I PENDAHULUAN

1.1 Latar Belakang

Air merupakan sumber daya yang sangat diperlukan makhluk hidup baik untuk memenuhi kebutuhan maupun menopang hidupnya secara alami. Kegunaan air yang bersifat universal atau menyeluruh dari setiap aspek kehidupan menjadi semakin berharganya air baik dilihat dari segi kuantitas maupun kualitasnya. Air pada batas tertentu sangat bermanfaat bagi kehidupan manusia, khususnya pada musim kemarau air sangatlah berarti, sebagian wilayah Indonesia yang mengalami kekeringan selalu kesulitan air. Jumlah wilayah yang menderita kekeringan dari tahun ketahun terlihat semakin meningkat dan meluas. Hal ini diakibatkan tidak hanya oleh rusaknya lingkungan di daerah tangkapan air akan tetapi juga diakibatkan oleh pesatnya pembangunan fisik serta rendahnya tingkat kesadaran masyarakat dalam penggunaan air tanpa diikuti dengan upaya menjaga dan melestarikan sumber daya air. Pada saat musim penghujan air sangat melimpah dan sangat mudah didapatkan sehingga banyak manusia yang justru boros dalam menggunakan air seperti yang terjadi pada pemakaian air di dalam gedung bertingkat, seiring dengan perkembangan teknologi maka dibuatlah suatu alat penampung air berupa tandon yang diletakkan di atas gedung bertingkat, dengan penerapan kontroler yang sesuai agar diketahui nilai perhitungan pemakaian daya elektrik pada sistem distribusi supply air ke dalam tandon dengan menggunakan pompa air disaat adanya pemakaian air didalam gedung bertingkat, baik secara berkala maupun terus menerus sehingga dapat menghemat pemakaian daya elektrik (Sudarmadi, 2011).

Oleh karena itu pada skripsi kali ini dibuatlah suatu alat yang membahas tentang pengontrolan level ketinggian air pada tandon untuk gedung bertingkat guna penghematan pemakaian daya elektrik dengan membandingkan penggunaan kontroler PID dan kontroler *on-off*, dimana pada sistem dengan menggunakan kontroler PID memiliki kesederhanaan struktur serta kemudahan dalam melakukan tuning parameter kontrolnya. Penentuan parameter yang sesuai agar mendapatkan respon keluaran sistem seperti pada yang diharapkan, untuk itu kontrol PID dapat dilakukan dengan metode *Ziegler Nichols* 1 untuk menentukan parameter K_p , K_i , K_d . Pada sistem menggunakan kontroler *on-off* hanya bekerja pada dua posisi, yaitu *on* dan *off*. Kerja kontroler *on-off* banyak digunakan pada aksi

pengontrolan yang sederhana karena harganya yang murah dan sistem kerja yang digunakan adalah *on-off* saja.

Tujuan dari penggunaan kontroler PID dan kontroler *on-off* untuk membandingkan penggunaan pemakaian daya elektrik dengan variasi beban maupun variasi beban menggunakan gangguan dengan besarnya nilai pada katub keluaran air yang sama dengan mengamati respon sistem, serta di harapkan mampu mengetahui perbedaan penghematan pemakaian daya elektrik dengan menggunakan kontroler PID dibandingkan dengan kontroler *on-off*.

1.2 Rumusan Masalah

Berdasarkan latar belakang dapat disusun rumusan masalah sebagai berikut:

1. Bagaimana perancangan algoritma dan performansi sistem pengontrolan level ketinggian air di dalam tandon dengan menggunakan kontroler PID dan kontroler *on-off*?
2. Bagaimana pemakaian daya elektrik pada proses pengontrolan level ketinggian air di dalam tandon dengan membandingkan penggunaan kontroler PID dan kontroler *on-off*?

1.3 Batasan Masalah

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

1. *Plant* dibuat dalam suatu modul berupa miniature tandon yang terbuat dari bahan *acrylic* dengan ukuran panjang 30 cm, lebar 4 cm dan tinggi 20 cm.
2. Mikrokontroler yang digunakan adalah arduino mega 2560.
3. Sensor yang digunakan adalah sensor tegangan, sensor arus dan sensor level ping ultrasonik yang berfungsi sebagai pendeteksi level ketinggian air.
4. Motor Pompa air DC mempunyai spesifikasi catu daya 12 V, arus 0,68 A dan debit maksimal 4,56 liter/menit.
5. *Plant* diberikan beban dan gangguan berupa nilai katub keluaran air yang berbeda.
6. Pembahasan ditekankan pada pemakaian daya elektrik dengan membandingkan penggunaan kontroler PID dan kontroler *on-off* pada prosed pengontrolan level ketinggian air di dalam tandon.

1.4 Tujuan

Terwujudnya suatu sistem pengontrolan level ketinggian air di dalam tandon dengan membandingkan penggunaan kontroler PID dan kontroler *on-off* guna mengetahui penghematan pemakaian daya elektrik.

1.5 Sistematika Pembahasan

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

BAB I PENDAHULUAN

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

BAB II TINJAUAN PUSTAKA

Menjelaskan dasar teori yang mendukung dalam pembuatan alat dan perancangan algoritma yang terdiri atas *Plant*, mikrokontroler, kontroler PID, metode *Ziegler-nichols*, kontroler *on-off*, motor pompa air DC, *driver* motor, sensor arus, sensor tegangan dan sensor ping ultrasonik.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan bagaimana melakukan pengujian alat-alat yang digunakan untuk membuat sistem pengontrolan level ketinggian air di dalam tandon dengan menggunakan kontroler PID dan kontroler *on-off* serta penentuan fungsi alih *Plant*, *driver*, sensor dan pembuatan algoritma.

BAB IV HASIL DAN PEMBAHASAN

Membahas hasil pengujian sistem, simulasi sistem dan analisis data secara keseluruhan terhadap alat yang telah direalisasikan.

BAB V KESIMPULAN DAN SARAN

Menjelaskan tentang pengambilan kesimpulan sesuai dengan hasil perancangan algoritma dan pengujian alat serta saran yang diperlukan untuk dilakukan pengembangan selanjutnya.



BAB II

TINJAUAN PUSTAKA

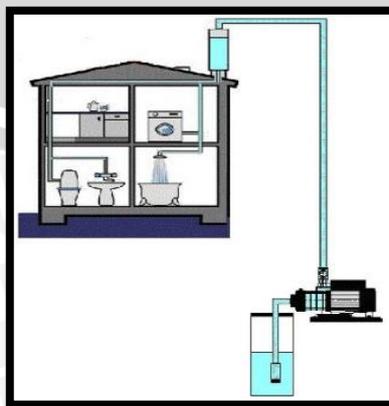
Untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan dari sistem yang di buat, maka perlu adanya penjelasan dan uraian mengenai teori penunjang yang digunakan dalam penulisan skripsi ini. Teori penunjang yang akan di bahas dalam bab ini adalah:

- *Plant* berupa tandon air.
- Mikrokontroler arduino Mega 2560.
- Kontroler PID
- Metode *ziegler nichols 1*.
- Kontroler *on-off*.
- Motor pompa air DC 12V.
- *Driver EMS 5A H-BRIDGE*.
- Sensor tegangan.
- Sensor arus ACS 712 5A
- Sensor ping ultrasonik.

2.1 *Plant*

Plant yang digunakan berupa tandon penampung air yang diletakkan di atas gedung bertingkat dengan mengontrol level ketinggian air agar diketahui pemakaian daya elektik pada proses suplai air ke dalam tandon dengan menggunakan motor pompa air DC 12 V dimana pada sistem di beri *input* berupa *setpoint* sebesar 15 cm berupa level ketinggian air.

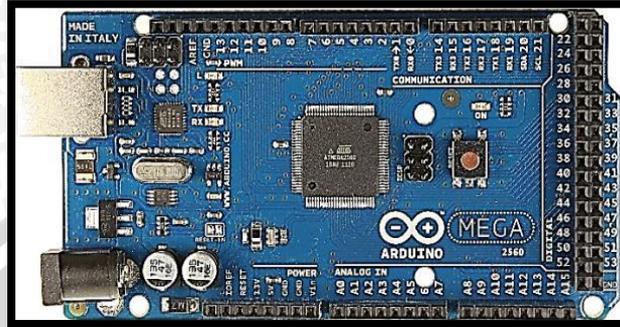
Plant dapat dilihat pada Gambar 2.1.



Gambar 2.1 Tandon Air

2.2 Mikrokontroler Arduino Mega 2560

Arduino Mega 2560 adalah papan mikrokontroler berdasarkan ATmega328 (lihat gambar 2.2). Board ini memiliki 54 pin digital *input/output* (14 pin dapat digunakan sebagai *output* PWM), 16 *input* analog, 16 MHz osilator kristal, USB koneksi, jack listrik, header ICSP, dan tombol reset. Mikrokontroler arduino mega 2560 dapat dilihat pada Gambar 2.2.



Gambar 2.2 Arduino Mega 2560
(Sumber : www.electroschematics.com)

2.2.1 Tegangan Sumber

Arduino Mega 2560 dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Eksternal (non-USB) daya dapat berasal baik dari AC ke adaptor DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan plug jack pusat-positif ukuran 2.1 mm konektor power. Ujung kepala dari baterai dapat dimasukkan kedalam Gnd dan Vin pin header dari konektor power. Arduino dapat beroperasi dengan catu daya eksternal 6 V sampai 20 V. Namun jika menggunakan lebih dari 12 V, regulator tegangan bisa panas dan merusak papan. Kisaran yang disarankan adalah 7 V sampai 12 V.

2.2.2 Memori

ATmega 2560 memiliki 256 KB (dengan 8 KB digunakan untuk bootloader), 8KB dari SRAM dan 4 KB EEPROM.

2.2.3 Input dan Output

Masing-masing dari 54 pin digital di Arduino Mega 2560 dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`, beroperasi dengan daya 5 V. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki internal *pull-up* resistor (secara *default* terputus) dari 20-50 Kohm. Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX); Serial 1: 19 (RX) dan 18 (TX); Serial 2: 17 (RX) dan 16 (TX); Serial 3: 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan

mengirimkan (TX) TTL data serial. Pin ini dihubungkan ke pin yang berkaitan dengan chip Serial ATmega8U2 USB-to-TTL.

- Eksternal *interrupts*: 2 (*interrupt* 0), 3 (*interrupt* 1), 18 (*interrupt* 5), 19 (*interrupt* 4), 20 (*interrupt* 3), dan 2 (*interrupt* 2). Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, dengan batasan tepi naik atau turun, atau perubahan nilai.
- PWM: 0 - 13. Menyediakan *output* PWM 8-bit dengan fungsi `analogWrite ()`.
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Pin ini mendukung komunikasi SPI menggunakan *SPI library*.
- LED: 13. Ada built-in LED terhubung ke pin digital 13. Ketika pin bernilai nilai high LED menyala dan ketika pin bernilai low LED mati.
- I2C: 20 (SDA) dan 21 (SCL). Dukungan I2C (TWI) komunikasi menggunakan *wire*.

2.2.4 Komunikasi

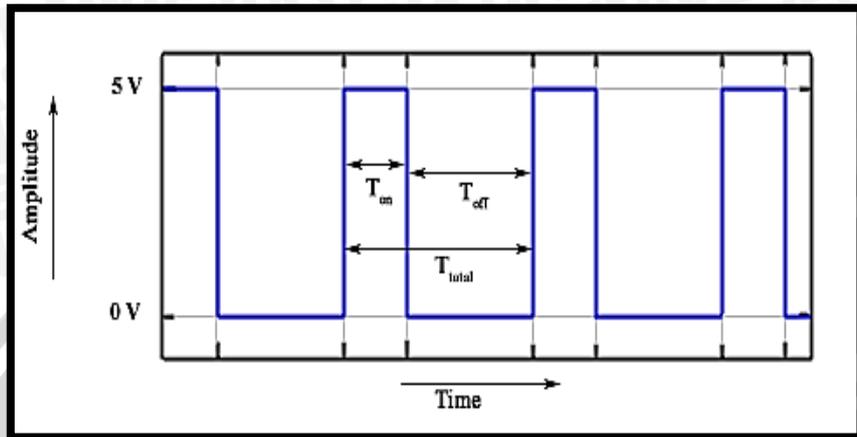
Arduino Mega 2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. Arduino Mega 2560 menyediakan 4 UART TTL (5V) untuk komunikasi serial. Sebuah Arduino Mega 2560 sebagai saluran komunikasi serial melalui USB dan sebagai port virtual com untuk perangkat lunak pada komputer. Firmware '8 U2 menggunakan *driver* USB standar COM, dan tidak ada *driver* eksternal yang diperlukan. Namun pada Windows diperlukan sebuah file inf. Perangkat lunak Arduino terdapat monitor serial yang memungkinkan digunakan memonitor data tekstual sederhana yang akan dikirim ke 8 atau dari papan Arduino. LED RX dan TX di papan tulis akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dengan koneksi USB ke komputer.

Sebuah *Software Serial Library* memungkinkan untuk berkomunikasi secara serial pada salah satu pin digital pada board Arduino Mega 2560. Arduino Mega 2560 juga mendukung I2C (TWI) dan komunikasi SPI. Perangkat lunak Arduino termasuk perpustakaan Kawat untuk menyederhanakan penggunaan bus I2C.

2.3 Pulse Width Modulation

Pengaturan tegangan sumber biasanya menggunakan metode *Pulse Width Modulation* (PWM). Sinyal *Pulse Width Modulation* (PWM) adalah metode yang dapat digunakan untuk mengontrol kecepatan motor DC. Dimana kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut.

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. *Duty cycle* adalah besarnya sinyal kontrol yang diberikan pada motor. Persamaan untuk perhitungan *duty cycle* ditunjukkan pada Persamaan 2.1 dengan T_{on} adalah periode logika tinggi, dan T adalah periode keseluruhan. Sinyal PWM secara umum dapat dilihat pada Gambar 2.3.



Gambar 2.3 Sinyal PWM secara umum

(Sumber: www.8051projects.net)

$$\text{Duty Cycle} = \frac{T_{on}}{T_{total}} \times 100\% \dots\dots\dots (2-1)$$

Sedangkan frekuensinya dapat ditentukan dengan Persamaan 2.2

$$f_{OCn} = \frac{f_{clk} / O}{N \cdot 256} \dots\dots\dots (2-2)$$

Timer atau *counter* yang digunakan pada PWM ini yaitu *timer* atau *counter* 0 (8 bit) dengan metode fast PWM dan *prescaler factor* (N) yaitu 256.

2.4 Kontroler PID (Proporsional Integral Diferensial)

Sistem pengontrolan yang di rancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengontrolan adalah harus stabil. Di samping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relative, yakni tolak ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikena gangguan (Ogata K.,1997).

Suatu sistem dikatakan stabil, jika di beri gangguan maka sistem tersebut akan kembali ke dalam *steady state* yaitu *output* berada dalam keadaan tetap seperti tidak ada gangguan. Sistem di katakana 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 responstgransien berupa suatu osilasi teredam sebelum mencapai steady state. Dalam sistem pengontrolan terdapat 2 macam loop.

1. Pengontrolan dengan loop terbuka

Sistem kontrol loop terbuka adalah sistem kontrol yang keluarannya tidak berpengaruh pada aksi pengontrolannya. Jadi pada sistem kontrol loop terbuka, keluaran tidak diukur atau diumpun balikan untuk dibandingkan dengan masukan.

2. Pengontrolan dengan loop tertutup

Sistem kontrol loop tertutup adalah sistem kontrol yang keluarannya mempunyai pengaruh langsung pada aksi pengontrolannya. Di sebut juga sistem yang menggunakan umpan balik untuk memperkecil kesalahan sistem.

2.4.1 Kontroler Proporsional

Kontroler proporsional memiliki keluaran yang sebanding dengan sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Untuk kontroler dengan aksi kontrol proporsional, hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ adalah:

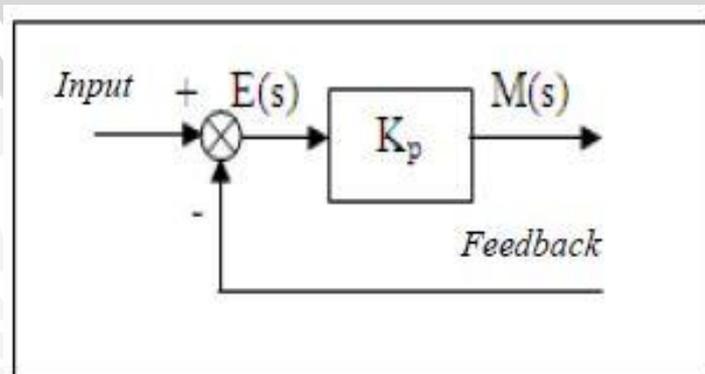
$$m(t) = K_p \cdot e(t) \dots\dots\dots (2-3)$$

atau, dalam besaran transformasi Laplace,

$$\frac{M(s)}{E(s)} = K_p \dots\dots\dots (2-4)$$

Di mana K_p adalah kepekaan proporsional atau penguatan.

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggeraknya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K.,1997). Diagram blok kontroler proporsional ditunjukkan pada Gambar 2.4



Gambar 2.4 Diagram Blok Kontroler Proporsional

(Sumber: Ogata K., 1997)



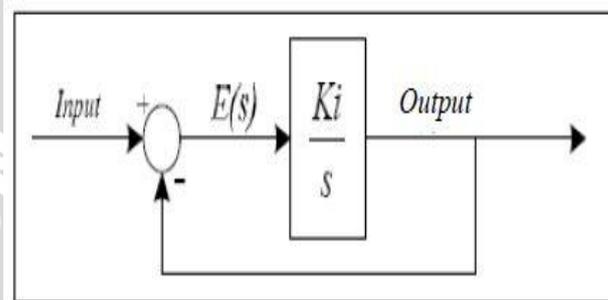
2.4.2 Kontroler Integral

Kontroler integral berfungsi mengurangi kesalahan keadaan mantap pada kontroler proporsional sebelumnya. Pada kontroler dengan aksi integral, harga keluaran kontroler $m(t)$ diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak $e(t)$.

Jadi,

$$\frac{dm(t)}{dt} = K_i \cdot e(t) \dots\dots\dots (2-5)$$

Jika harga $e(t)$ diduakalikan, maka harga $m(t)$ berubah dengan laju perubahan menjadi dua kali semula. Jika kesalahan penggerak nol, maka harga $m(t)$ tetap stasioner. Aksi kontrol integral seringkali disebut *control reset* (Ogata K.,1997). Diagram blok kontroler integral ditunjukkan pada Gambar 2.5



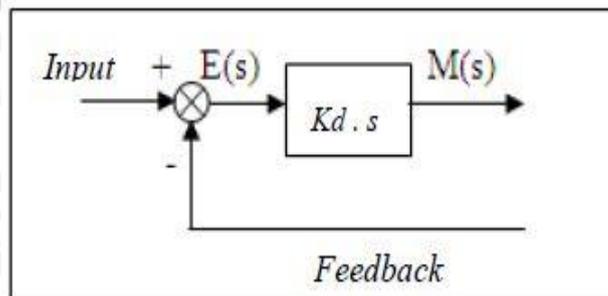
Gambar 2.5 Diagram Blok Kontroler Integral
(Sumber: Ogata K., 1997)

2.4.3 Kontroler Diferensial

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

$$\frac{M(s)}{E(s)} = K_d \cdot s \dots\dots\dots (2-6)$$

Diagram blok kontroler diferensial ditunjukkan pada Gambar 2.6



Gambar 2.6 Diagram Blok Kontroler Diferensial
(Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997)



2.4.4 Kontroler Proporsional Integral Diferensial (PID)

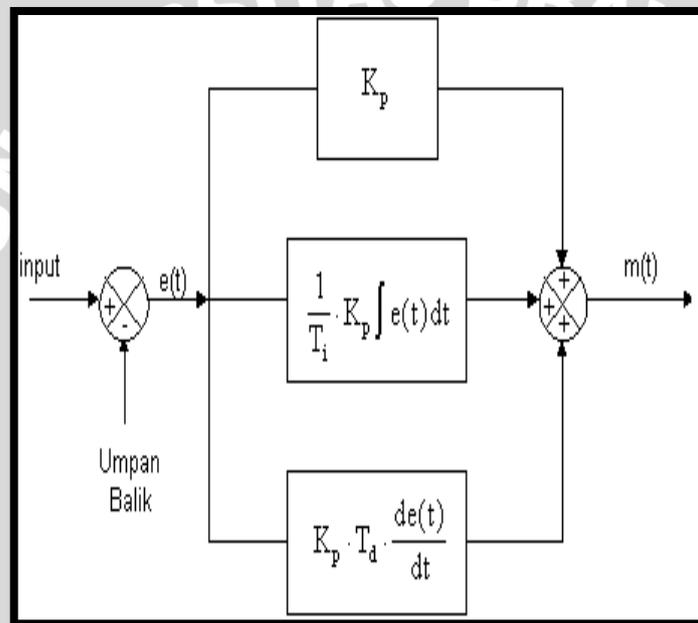
Gabungan aksi kontrol proporsional, integral, dan diferensial mempunyai keunggulan dapat saling menutupi kekurangan dan kelebihan dari masing-masing kontroler. Persamaan kontroler PID ini dapat dinyatakan sebagai berikut:

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \dots\dots\dots (2-5)$$

Dalam transformasi Laplace dinyatakan sebagai berikut:

$$\frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \dots\dots\dots (2-6)$$

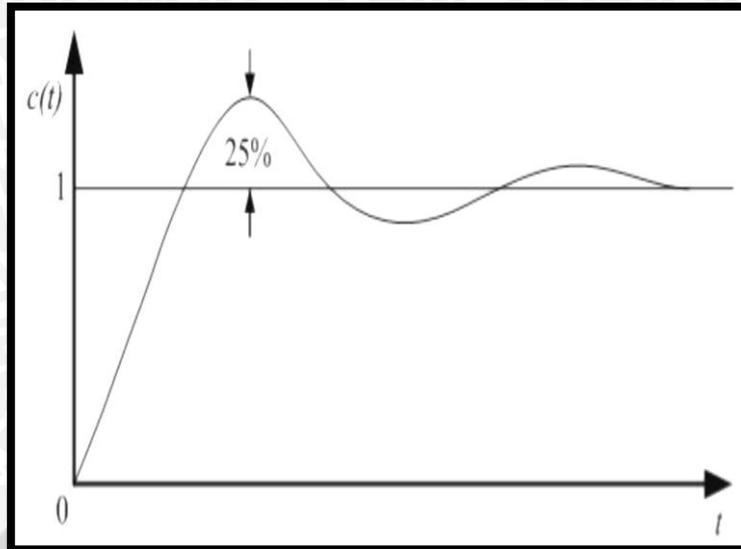
Dari rumus diatas maka didapatkan diagram blok seperti yang ditunjukkan pada gambar 2.7



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

2.5 Metode Ziegler-nichols

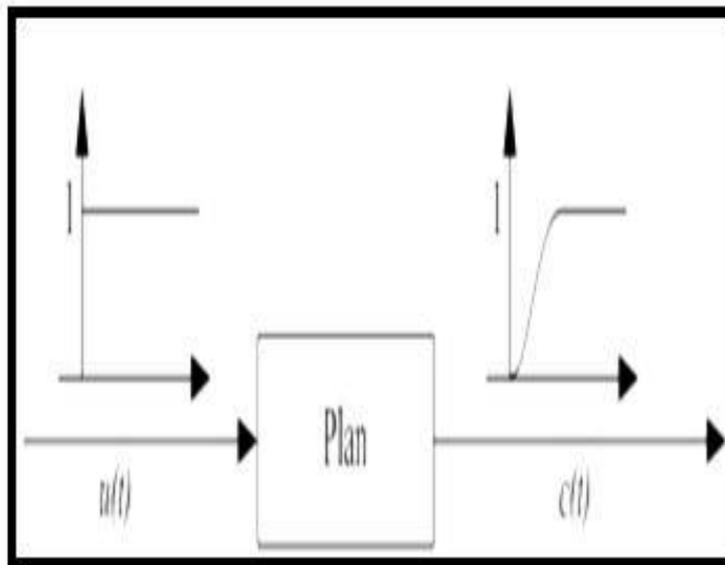
Ziegler dan Nichols mengemukakan aturan-aturan untuk menentukan nilai dari *gain* proporsional K_p , waktu integral T_i , dan waktu derivatif T_d berdasarkan karakteristik respon transien dari *Plant* yang diberikan. Penentuan parameter kontroler PID atau penalaan kontroler PID tersebut dapat dilakukan 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 yang ditunjukkan pada Gambar 2.8



Gambar 2.8 Kurva Respon Unit Step yang Menunjukkan 25% overshoot
(Sumber: Ogata, K., 1997)

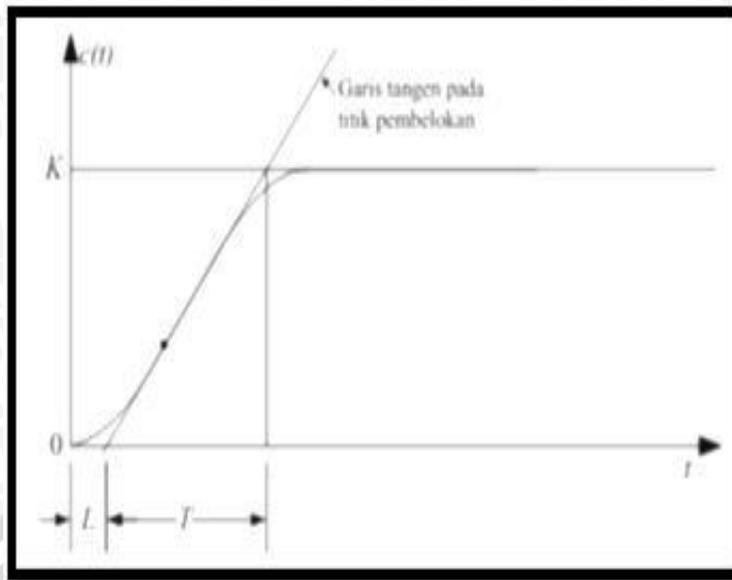
2.5.1 Metode Pertama

Metode pertama atau sering disebut metode kurva reaksi, respon dari plan dapat diperoleh secara eksperimental dengan masukan berupa unit step, seperti yang ditunjukkan pada Gambar 2.9



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

Jika dalam *Plant* tersebut terdapat integrator atau *dominan complex-conjugate poles*, maka kurva respon unit step berbentuk seperti huruf S, seperti yang ditunjukkan pada Gambar 2.10 (Ogata, K., 1997).



Gambar 2.10 Kurva Respon yang Berbentuk S
(Sumber: Ogata, K. 1997)

Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda L dan konstanta waktu T .

Waktu tunda dan konstanta waktu ditentukan dengan menggambar sebuah garis tangen pada titik pembelokan dari kurva S, dan menentukan perpotongan antara garis tangen dengan sumbu waktu t dan sumbu $c(t) = K$, seperti yang telah ditunjukkan dalam Persamaan sebagai berikut:

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts+1} \dots\dots\dots (2-7)$$

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

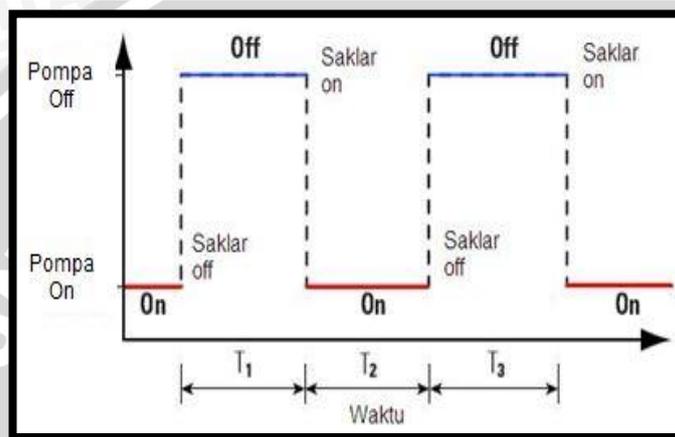
Tipe Kontroler	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	$2L$	$0,5L$

Tabel 2.1 Aturan Penalaan Ziegler-nichols Berdasarkan Respon Unit Step Dari Plant
(Sumber: Ogata, K. 1997)



2.6 Kontroler On-Off

Karakteristik kontroler *on – off* ini hanya bekerja pada 2 posisi, yaitu *on* dan *off*. Kerja kontroler *on – off* banyak digunakan pada aksi pengontrolan yang sederhana karena harganya murah. Karena sistem kerja yang digunakan adalah *on – off* saja, hasil *output* dari sistem pengendalian ini akan menyebabkan proses variabel tidak akan pernah konstan. Besar kecilnya fluktuasi process variabel ditentukan oleh titik dimana kontroler dalam keadaan *on* dan *off*. Pengendalian dengan aksi kontrol ini juga menggunakan *feedback*, seperti yang ditunjukkan pada Gambar 2.11.



Gambar 2.11 Aksi Kendali On-Off
(Sumber: www.repository.com)

Dari gambar 2.11 dapat diamati bahwa jika *output* lebih besar dari setpoint, aktuator akan *off*. *Output* akan turun dengan sendirinya sehingga menyentuh *setpoint* lagi. Pada saat itu, sinyal kontrol akan kembali *on* (aktuator *on*) dan mengembalikan *output* kepada *setpoint*. Demikian seterusnya sinyal kontrol dan aktuator akan *on-off* terus menerus. Kelemahan dari kontroler *on-off* ini adalah jika *output* beresilasi di sekitar *setpoint* (keadaan yang memang diinginkan) akan menyebabkan aktuator bekerja keras untuk *on-off* dengan frekuensi yang tinggi. Hal ini akan menyebabkan kontroler akan cepat aus dan memakan energi yang banyak (boros).

2.7 Pompa Air DC

Pompa adalah salah satu peralatan yang dipakai untuk mengubah energi mekanik (dari mesin penggerak pompa) menjadi energi tekan pada fluida yang dipompa. Pada umumnya pompa digunakan untuk memindahkan fluida dari suatu tempat ke tempat lain yang lebih tinggi tempatnya, tinggi tekanannya, ataupun untuk sirkulasi (Sunarno, Mekanikal Elektrikal). Pada sistem ini aktuator yang di gunakan adalah pompa air DC.

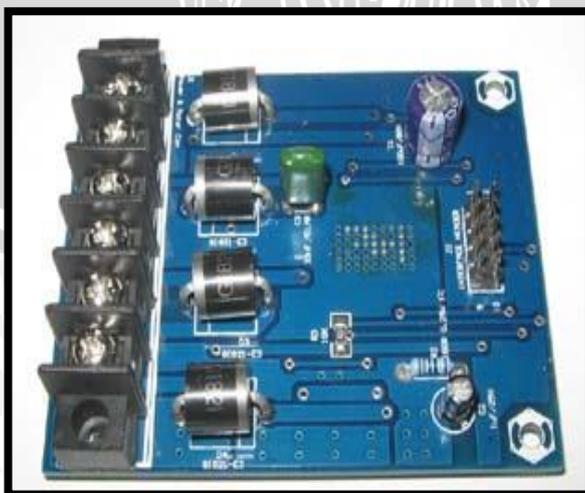
Fungsi dari pompa air DC ini adalah untuk mengalirkan air ke dalam tandon penampung air di atas gedung bertingkat. Pompa air yang digunakan sebagai aktuator ini memiliki tegangan masukan sebesar 12V DC, arus 0.68 A. Daya hisap yang dihasilkan pompa ini adalah sebesar 4.56 liter per menit seperti yang ditunjukkan pada gambar 2.12.



Gambar 2.12 Pompa Air DC 12 V
(Sumber: Yoonitania, 2013)

2.8 Driver EMS 5A H-BRIDGE

Embedded Module Series (EMS) 5A H-Bridge merupakan *driver* H-Bridge yang didisain untuk menghasilkan drive 2 arah dengan arus kontinyu sampai dengan 5A pada tegangan 5 Volt sampai 40 Volt. Modul ini dilengkapi dengan rangkaian sensor arus beban yang dapat digunakan sebagai umpan balik ke pengendali. Modul ini mampu men-drive beban-beban induktif seperti misalnya relay, solenoida, motor DC, motor stepper, dan berbagai macam beban lainnya, seperti yang ditunjukkan pada Gambar 2.13.



Gambar 2.13 Motor Driver EMS 5A H-Bridge
Sumber: (www.sir.stikom.edu.com)

➤ Spesifikasi:

1. Terdiri dari 1 *driver* full H-Bridge beserta rangkaian current sense.
2. Mampu melewatkan arus kontinyu 5 A.
3. Range tegangan *output* untuk beban: 5 V sampai 40 V.
4. *Input* kompatibel dengan level tegangan TTL dan CMOS.
5. Jalur catu daya *input* (VCC) terpisah dari jalur catu daya untuk beban (V Mot).
6. *Output* tri-state.
7. Dilengkapi dengan dioda eksternal untuk pengaman beban induktif.
8. Frekuensi PWM sampai dengan 10 KHz. 9. Active Current Limiting.

2.8.1 Keterangan *Driver* EMS 5A H-BRIDGE

Modul H-Bridge memiliki 1 set header (J2) dan 1 set terminal konektor (J1). Pada bagian ini akan dijelaskan deskripsi dan fungsi dari masing-masing header dan konektor tersebut. Interface Header (J2) berfungsi sebagai *input* untuk antarmuka dengan *input-output* digital serta *output* analog dari modul H-Bridge. Berikut deskripsi dari masing-masing pin pada Interface Header, seperti yang ditunjukkan pada Tabel 2.2.

No. Pin	Nama	I/O	Fungsi
1	MIN1	I	Pin input untuk menentukan output MOUT 1
2	MIN2	I	Pin input untuk menentukan output MOUT 2
3	MSTAT1	O	Output digital yang melaporkan adanya kondisi <i>fault</i> pada modul. Berlogika Low jika ada <i>fault</i> pada modul atau output
4	MEN	I	Pin <i>enable</i> untuk output H-Bridge (MOUT 1 dan MOUT 2)
5	MCS	O	Output tegangan analog yang berbanding lurus dengan arus beban (Range output 0 – 2,5 Volt)
6	MSLP	I	Pin input untuk mengatur kerja modul H-Bridge. Diberi logika High untuk <i>Full Operation</i> , diberi logika Low untuk <i>Mode Sleep</i>
7,9	VCC	-	Terhubung ke catu daya untuk input (5 Volt)
8,10	PGND	-	Titik referensi untuk catu daya input

Tabel 2.2 Motor *Driver* EMS 5A H-Bridge
(Sumber: *Datasheet Driver* EMS 5A H-Bridge)

Arus (dalam *Ampere*) yang dilewatkan oleh *H-Bridge* dapat dihitung dengan rumus:

$$I = \frac{\text{Tegangan_output_pada_pin_MCS}}{180} \times 375$$

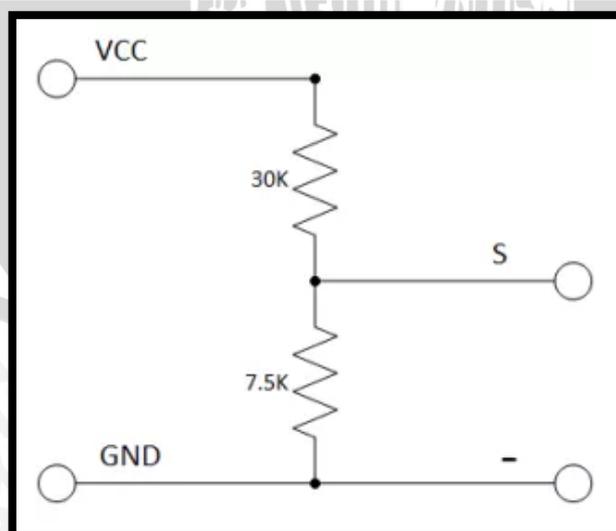
Power & Motor Con (J1) berfungsi sebagai konektor untuk catu daya dan beban. Berikut deskripsi dari masing-masing terminal pada Power & Motor Con, seperti yang ditunjukkan pada Tabel 2.3

Nama	Fungsi
PGND	Titik referensi untuk catu daya input
VCC	Terhubung ke catu daya untuk input (5 Volt)
MGND	Titik referensi untuk catu daya output ke beban
V MOTOR (V MOT)	Terhubung ke catu daya untuk output ke beban
MOUT 2	Output ke beban dari <i>half</i> H-Bridge kedua
MOUT 1	Output ke beban dari <i>half</i> H-Bridge pertama

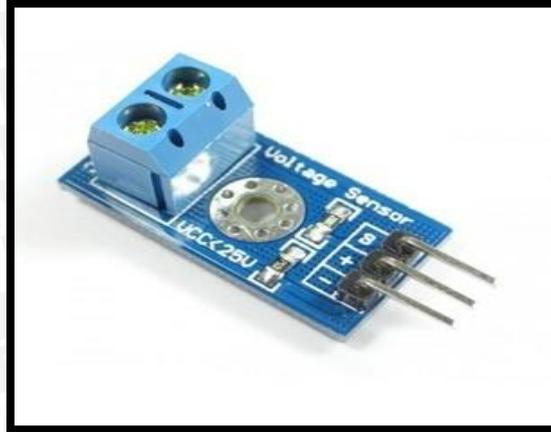
Tabel 2.3 Deskripsi Power & Motor Con pada masing-masing terminal
(Sumber: *Datasheet Driver EMS 5A H-Bridge*)

2.9 Sensor Tegangan

Sensor tegangan merupakan sebuah *device* yang digunakan mengukur tegangan pada nilai rentang tertentu yang nantinya data pengukuran diolah dalam bentuk digital. Jenis sensor tegangan jika dibagi skala penggunaan ada dua yaitu dunia elektronika dan dunia industry. Untuk dunia elektronika sendiri terdapat modul sensor tegangan robot Arduino good (Gambar 2.17). Cara kerja sensor ini berpacu pada kerja board mikrokontroler Arduino, diketahui *input* analog Arduino terbatas pada 5V C. Jika ingin mengukur tegangan lebih tinggi dari nilai tersebut dibutuhkan alat tambahan agar tidak merusak board. Alat tambahan tersebut adalah sensor tegangan, pada dasarnya adalah pembagian tegangan menggunakan resistor 30k Ω dan 7,5k Ω karena perbandingannya 5:1 jadi batas maksimal dari sensor ini adalah 25 V DC, seperti yang ditunjukkan pada Gambar 2.14 dan 2.15.



Gambar 2.14 Rangkaian Ekuivalen
(Sumber: www.henrysbench.capnfatz.com)



Gambar 2.15 Sensor Tegangan
(Sumber: www.henrysbench.capnfatz.com)

2.10 Sensor Arus

Sensor ACS 712 adalah merupakan sensor untuk mendeteksi arus. Penggunaan sensor arus ACS 712 ini Kebanyakan memiliki kekurangan yakni nilai arus yang di dapatkan dari sensor tidak linear sehingga terkadang kita membutuhkan tingkat linear yang lebih tinggi. Sebelum membahas lebih lanjut, akan di jelaskan terlebih dahulu tentang sensor arus ACS 712. ACS 712 ini memiliki tipe variasi sesuai dengan arus maksimal yakni 5A, 20A, 30A. ACS 712 ini menggunakan VCC 5V, seperti yang ditunjukkan pada Gambar 2.16.



Gambar 2.16 Sensor Arus ACS 712 5A
(Sumber: www.jawaplugin.blogspot.co.id)

Pada pembahasan kali ini sensor arus yang di gunakan adalah variasi ACS712 dengan nilai arus maksimal 5A buatan LC Electronics dengan menggunakan Arduino MEGA 2560 sebagai Microcontrollernya. Sensor Arus ACS712 - 5A ini memang memiliki karakteristik dapat dilihat dari cuplikan dari *datasheet* ACS712 - 5A, seperti yang ditunjukkan pada Tabel 2.4.

x05B PERFORMANCE CHARACTERISTICS ¹ $T_A = -40^\circ\text{C}$ to 85°C , $C_F = 1\text{ nF}$, and $V_{CC} = 5\text{ V}$, unless otherwise specified						
Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_P		-5	-	5	A
Sensitivity	Sens	Over full range of I_P , $T_A = 25^\circ\text{C}$	180	185	190	mV/A
Noise	$V_{\text{NOISE(PP)}}$	Peak-to-peak, $T_A = 25^\circ\text{C}$, 185 mV/A programmed Sensitivity, $C_F = 47\text{ nF}$, $C_{\text{OUT}} = \text{open}$, 2 kHz bandwidth	-	21	-	mV
Zero Current Output Slope	$\Delta V_{\text{OUT(Q)}}$	$T_A = -40^\circ\text{C}$ to 25°C	-	-0.26	-	mV/°C
		$T_A = 25^\circ\text{C}$ to 150°C	-	-0.08	-	mV/°C
Sensitivity Slope	ΔSens	$T_A = -40^\circ\text{C}$ to 25°C	-	0.054	-	mV/A/°C
		$T_A = 25^\circ\text{C}$ to 150°C	-	-0.008	-	mV/A/°C
Total Output Error ²	E_{TOT}	$I_P = \pm 5\text{ A}$, $T_A = 25^\circ\text{C}$	-	± 1.5	-	%

¹Device may be operated at higher primary current levels, I_P , and ambient temperatures, T_A , provided that the Maximum Junction Temperature, $T_{J(\text{max})}$, is not exceeded.

²Percentage of I_P , with $I_P = 5\text{ A}$. Output filtered.

Gambar 2.4 Karakteristik Sensor Arus ACS 712 5A
(Sumber: *Datasheet* Sensor Arus ACS 712 5A)

2.11 Sensor Ping Ultrasonik

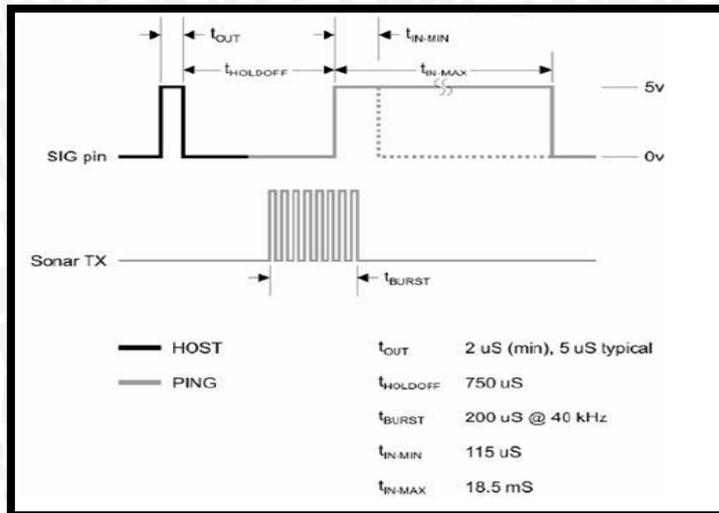
Sensor jarak ultrasonik PING adalah sensor 40 KHz produksi parallax yang banyak digunakan untuk aplikasi atau kontes robot cerdas untuk mendeteksi jarak suatu objek, seperti yang ditunjukkan pada Gambar 2.17.



Gambar 2.17 Sensor Jarak Ultrasonik PING
(Sumber: *Datasheet* Sensor Jarak Ultrasonik PING)

Sensor PING mendeteksi jarak objek dengan cara memancarkan gelombang ultrasonik (40 KHz) selama $t = 200\text{ us}$ kemudian mendeteksi pantulannya. Sensor PING memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroler pengendali (pulsa trigger dengan tout min 2 us). Spesifikasi sensor ini:

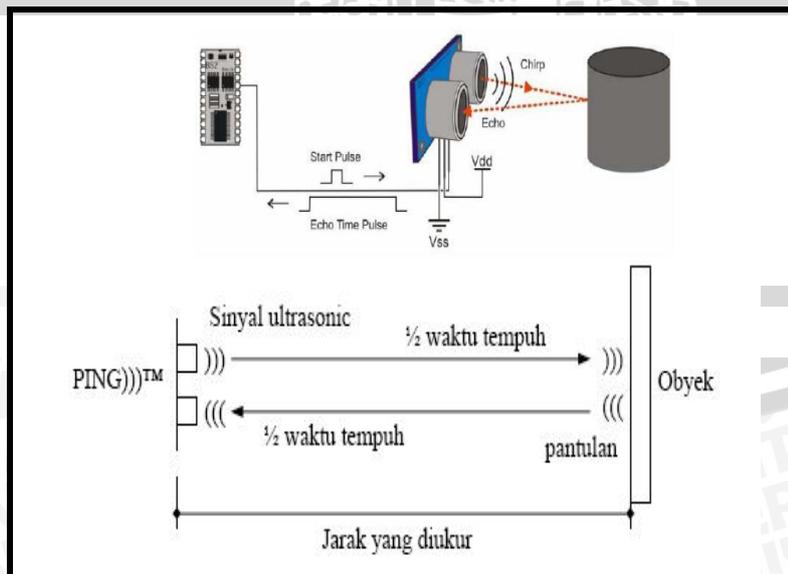
- Kisaran pengukuran 3cm-3m.
- Input trigger –positive TTL pulse, 2uS min., 5uS tipikal.
- Echo hold off 750uS dari fall of trigger pulse.
- Delay before next measurement 200uS.
- Burst indicator LED menampilkan aktifitas sensor.



Gambar 2.18 Diagram Waktu Sensor Ping Ultrasonik
(Sumber: *Datasheet* Sensor Ping Ultrasonik)

2.11.1 Prinsip Kerja Sensor Ping Ultrasonik

Pada dasarnya, Sensor PING terdiri dari sebuah chip pembangkit sinyal 40KHz, sebuah speaker ultrasonik dan sebuah mikropon ultrasonik. Speaker ultrasonik mengubah sinyal 40 KHz menjadi suara sementara mikropon ultrasonik berfungsi untuk mendeteksi pantulan suaranya. Sensor PING mendeteksi jarak obyek dengan cara memancarkan gelombang ultrasonik (40 kHz) selama t_{BURST} (200 μ s) kemudian mendeteksi pantulannya. Sensor PING memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroler pengendali (pulsa *trigger* dengan t_{OUT} min. 2 μ s), seperti yang ditunjukkan pada Gambar 2.19.



Gambar 2.19 Prinsip Kerja Sensor Ping Ultrasonik
(Sumber: *Datasheet* Sensor Ping Ultrasonik)

Gelombang ultrasonik ini melalui udara dengan kecepatan 344 meter per detik, mengenai obyek dan memantul kembali ke sensor. PING mengeluarkan pulsa *output high* pada pin SIG setelah memancarkan gelombang ultrasonik dan setelah gelombang pantulan terdeteksi PING akan membuat *output low* pada pin SIG. Lebar pulsa High (tIN) akan sesuai dengan lama waktu tempuh gelombang ultrasonik untuk 2x jarak ukur dengan obyek. Maka jarak yang diukur adalah

$$S = (tIN \times V) \div 2$$

Dimana:

S = Jarak antara sensor ultrasonik dengan objek yang dideteksi.

V = Cepat rambat gelombang ultrasonik di udara (344 m/s).

tIN = Selisih waktu pemancaran dan penerimaan pantulan gelombang.

2.12 Debit Air

Debit air adalah banyaknya volume zat cair yang mengalir pada tiap satu satuan waktu, biasanya dinyatakan dalam satuan liter/detik atau dalam satuan meter kubik (m³) per detik.

Rumus menghitung debit air:

$$Q = A \cdot v \text{ atau } Q = \frac{V}{t}$$

Keterangan:

Q = debit (m³/s).

V = volume fluida (m³).

t = waktu (s).

A = luas (m²).

v = kecepatan aliran (m/s).

$v = \sqrt{2gh}$.

g = gravitasi.

h = jarak permukaan cairan ke lubang bocor.

2.13 Daya Elektrik DC

Daya adalah laju energi yang dihantarkan atau kerja yang dilakukan per satuan waktu. Daya dilambangkan dengan P. Untuk menghitung daya listrik oleh arus bolak balik (AC) dengan arus searah DC itu berbeda, karena arus AC mempunyai frekuensi sehingga

memiliki perhitungan tersendiri. Sedangkan untuk arus DC yang melewati resistor untuk menghitung daya yang dihasilkannya dapat menggunakan rumus :

$$P = V \times I$$

Keterangan:

P = daya listrik dalam satuan Watt.

V = tegangan listrik dalam satuan Volt (V).

I = Arus listrik dalam satuan Ampere (A).



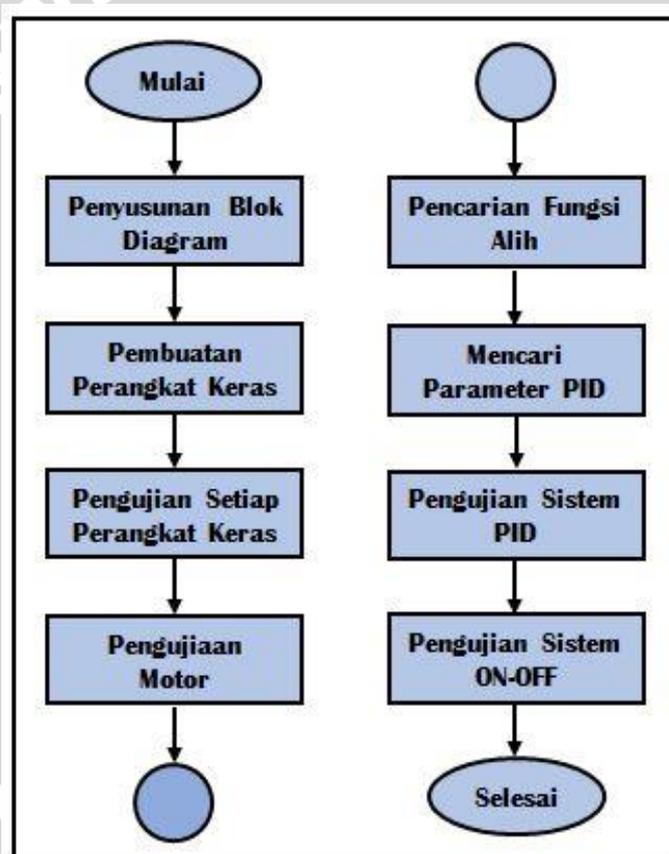
BAB III METODOLOGI PENELITIAN

3.1 Metode Penelitian

Kajian yang digunakan dalam skripsi kali ini adalah kajian tentang penelitian mengenai pengontrolan level ketinggian air pada tandon untuk gedung bertingkat guna membandingkan respon sistem menggunakan kontroler PID dengan *on-off* pada pemakaian daya elektrik DC.

3.1.1 Kerangka Penelitian

Berikut ini adalah kerangka penelitian yang dapat dilihat pada gambar 3.1



Gambar 3.1 Kerangka Penelitian

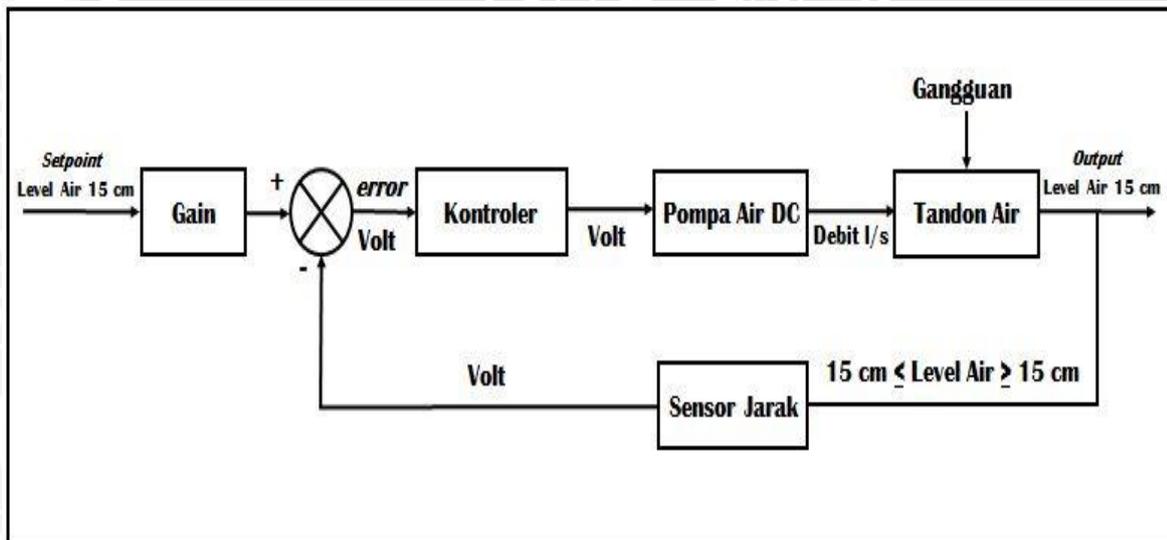
Adapun penjelasan dari *flowchart* sebagai berikut :

1. Penyusunan blok diagram sistem bertujuan untuk memahami struktur sistem pada setiap bloknnya sehingga akan mempermudah dalam analisis secara keseluruhan.

2. Pembuatan perangkat keras bertujuan mengetahui kebutuhan perangkat keras sistem yang dibutuhkan serta mengetahui cara kerja sistem dari sisi perangkat keras.
3. Pengujian setiap perangkat keras bertujuan untuk mengetahui karakteristik dan fungsi alihnya.
4. Pengujian motor bertujuan mengetahui hubungan tegangan masukan dengan putaran motor.
5. Pencarian fungsi alih bertujuan untuk mengetahui karakteristik sistem agar dapat dicari parameter kontrolernya.
6. Mencari parameter kontrol PID menggunakan metode *Ziegler-nichols* 1 agar sistem dapat bekerja dengan baik yaitu keluaran diharapkan sama dengan masukan yang diinginkan.
7. Pengujian sistem bertujuan untuk mengetahui apakah sistem dapat bekerja dengan baik atau tidak.

3.2 Penyusunan Blok Diagram Sistem

Pada perencanaan alat diperlukan perencanaan blok diagram sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai rencana, berikut blok diagram sistem, seperti yang ditunjukkan pada Gambar 3.2.



Gambar 3.2 Diagram Blok Sistem

Keterangan dari diagram blok sistem dalam Gambar 3.2 adalah sebagai berikut:

- *Setpoint* sistem adalah level ketinggian air sebesar 15 cm dan debit air yang dihasilkan oleh karakteristik pompa air DC 12V sebesar 4.56 liter/menit.

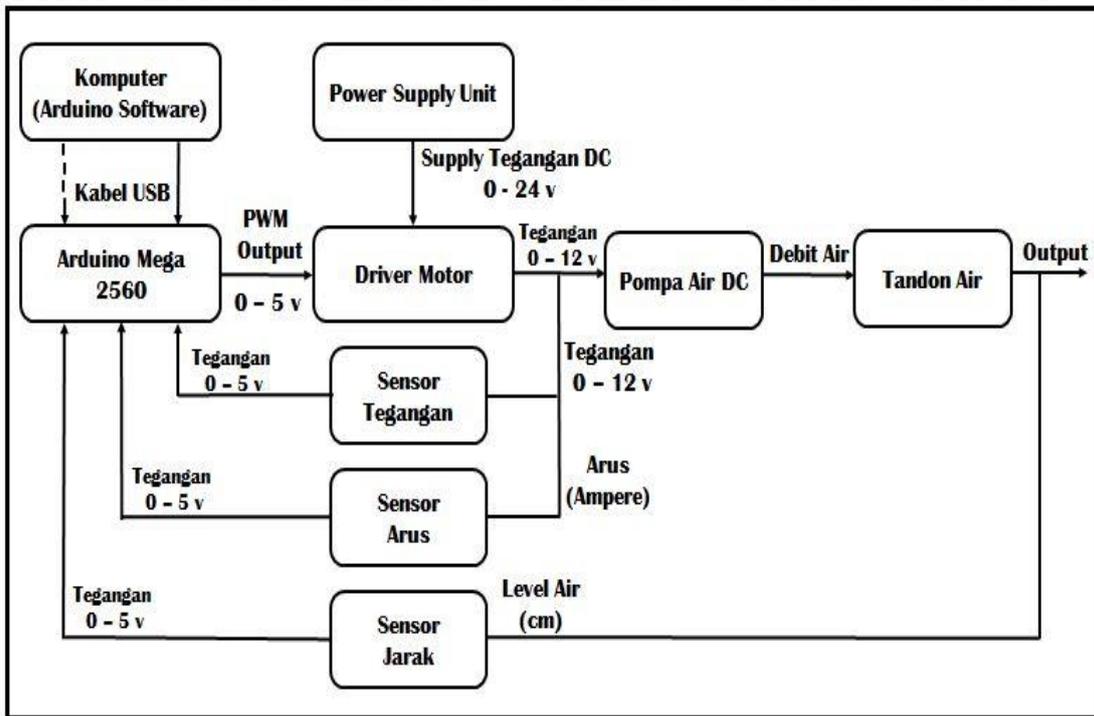
- Tandon dilengkapi dengan satu katub keluaran air terbuka dan enam katub keluaran air (buka tutup) dibawah tangki sebagai gangguan pada pemakaiian air di dalam gedung bertingkat yang digunakan untuk variasi nilai katub keluaran air atau beban.
- Pusat pengendalian sistem adalah Arduino Mega 2560 yang memberikan sinyal PWM pada pompa melalui *driver* EMS H-Bridge 5A.
- Aktuator menggunakan satu motor pompa air DC 12V yang berfungsi untuk mengalirkan air ke pipa menuju tandon penyimpanan yang mendapat sinyal masukan dari *output* PWM *driver* EMS H-Bridge 5A.
- Sensor jarak menggunakan ping ultrasonik yang berfungsi sebagai *feedback* (umpan balik) agar level ketinggian air sesuai dengan *setpoint* yang diinginkan dengan jarak rentang 0-15 cm.
- Kontroler yang digunakan adalah Kontrol PID dan kontroler *on-off* dengan menggunakan Arduino Mega 2560, dimana pada kontroler PID menggunakan metode *Ziegler-nichols* 1 untuk mendapatkan parameter kontrolnya.

3.3 Pembuatan Perangkat Keras

Pembuatan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta pemrogramannya, hal ini dimaksudkan agar sistem pengontrolan level ketinggian air dapat diketahui penggunaan pemakaiian daya elektrik seperti apa yang telah direncanakan.

Pembuatan perangkat keras meliputi:

1. Skema pembuatan perangkat keras (Gambar 3.3)
2. Penentuan modul elektronik yang digunakan, meliputi :
 - Power Supply Unit.
 - Mikrokontroler Arduino Mega 2560.
 - *Driver* Motor EMS H-Bridge 5A.
 - Sensor Arus ACS 712 5A.
 - Sensor Tegangan.
 - Sensor Jarak (Ping Ultrasonik).
 - Motor pompa air DC 12V.
 - Tandon Penampung Air.



Gambar 3.3 Skema Pembuatan Perangkat Keras

3.4 Spesifikasi Alat

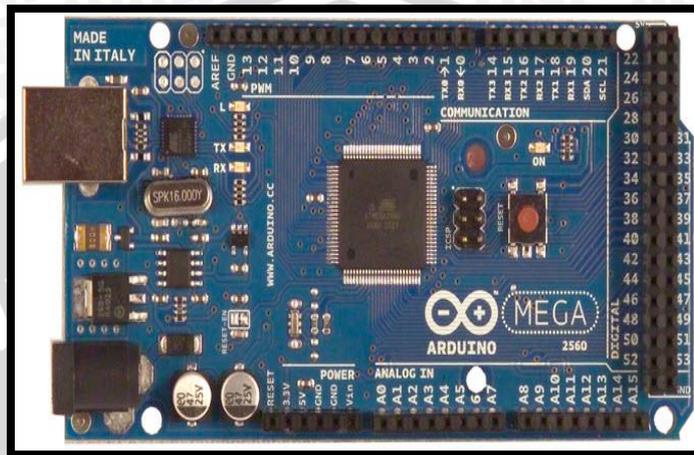
Spesifikasi alat yang meliputi komponen-komponen pendukung pada skripsi ini adalah sebagai berikut:

1. Komputer atau PC yang sudah terinstall *software* Arduino dan MATLAB 2014a.
2. *Power Supply Unit* (PSU), berfungsi sebagai sumber tegangan atau catu daya pada *driver* motor dimana PSU yang digunakan ini mempunyai tegangan keluaran 0-12V, seperti yang ditunjukkan pada Gambar 3.4.



Gambar 3.4 *Power Supply Unit*

3. Perangkat kontroler yang digunakan adalah Arduino Mega 2560. Mikrokontroler Arduino Mega 2560 berfungsi sebagai penyimpan algoritma kontroler sistem, pengendali *driver* motor, dan menerima serta mengolah data yang dikirimkan sensor ping ultrasonik (sensor jarak). Mikrokontroler menggunakan catu tegangan sebesar 5 Volt yang didapat dari komputer atau PC, seperti yang ditunjukkan pada Gambar 3.5.



Gambar 3.5 Mikrokontroler Arduino Mega 2560

4. *Driver* motor yang digunakan adalah modul yang menggunakan EMS 5A *H-Bridge*. *Driver* motor berfungsi sebagai pengali tegangan dimana tegangan masukan dari mikrokontroler Arduino dikali dengan sebuah nilai *gain* atau penguatan tertentu menjadi tegangan keluaran *driver* motor yang digunakan untuk mencatu motor. Catu tegangan *driver* motor didapat dari *power supply unit* (PSU) dengan tegangan maksimal 40 V dan arus maksimalnya sebesar 5 A, seperti yang ditunjukkan pada Gambar 3.6.



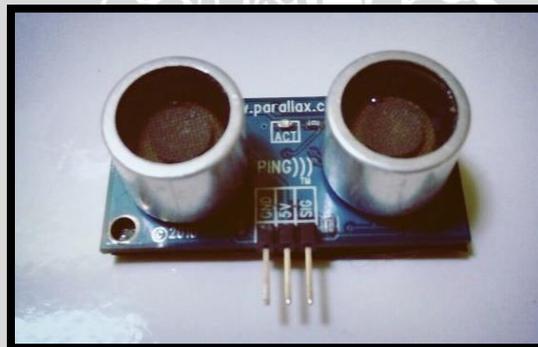
Gambar 3.6 Driver Motor EMS 5A H-Bridge

5. Motor pompa air DC yang digunakan memiliki tegangan maksimal sebesar 12 V, dan memiliki arus maksimal sebesar 0.68 A serta memiliki debit air sebesar 4.56 liter/menit yang berfungsi untuk mengalirkan air ke dalam tandon penyimpanan air, seperti yang ditunjukkan pada Gambar 3.7.



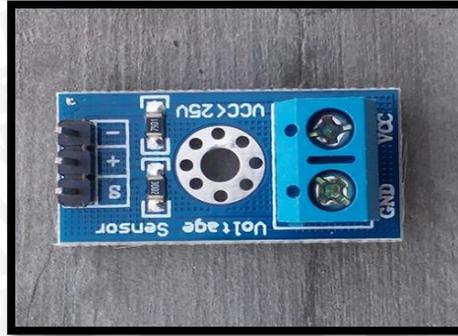
Gambar 3.7 Motor Pompa Air DC

6. Sensor jarak yang digunakan adalah ping ultrasonik, sensor ping ultrasonik membutuhkan suplai daya sebesar 5 V yang diperoleh dari mikrokontroler. Keluaran sensor ping ultrasonik berupa pulsa yang didapat dari proses pembacaan ketinggian yang kemudian dikirim ke mikrokontroler untuk diproses selanjutnya oleh Arduino Mega 2560, seperti yang ditunjukkan pada Gambar 3.8.



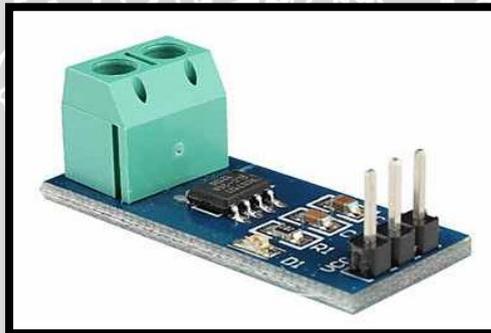
Gambar 3.8 Sensor Ping Ultrasonik

7. Sensor tegangan digunakan untuk pembacaan tegangan pada saat adanya catu daya dari sumber dengan *input* sebesar 0-25V. Sensor tegangan membutuhkan suplai daya sebesar 5 V yang diperoleh dari mikrokontroler. Keluaran sensor tegangan dihubungkan dengan rangkaian penguat agar dapat dibaca oleh mikrokontroler arduino mega 2560 dengan rentang 0-5V, seperti yang ditunjukkan pada Gambar 3.9.



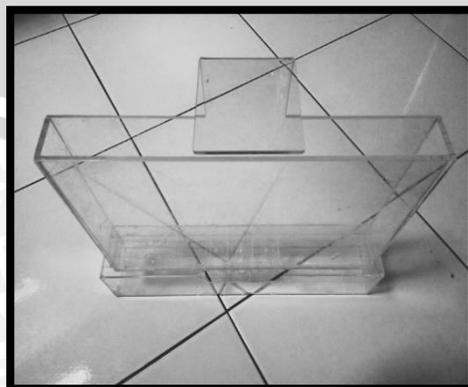
Gambar 3.9 Sensor Tegangan

8. Sensor arus ACS 712 5A dihubungkan dengan rangkaian kapasitor yang berguna sebagai filter serta pada pemasangan sensor arus ACS 712 5A harus disertai dengan beban motor pompa air DC. Sensor arus memiliki range masukan -5-5A, sensor arus juga membutuhkan suplai daya sebesar 5 V yang diperoleh dari mikrokontroler. Keluaran sensor arus dihubungkan dengan rangkaian penguat agar dapat dibaca oleh mikrokontroler arduino mega 2560 dengan rentang 0-5V, seperti yang ditunjukkan pada Gambar 3.10.



Gambar 3.10 Sensor Arus ACS 712 5A

9. *Plant* berupa tandon penampungan air yang digunakan terbuat dari bahan *acrylic* dengan ukuran panjang 30 cm, lebar 4 cm dan tinggi 20 cm, seperti yang ditunjukkan pada Gambar 3.11.



Gambar 3.11 *Plant* Berupa Tandon Air

3.5 Prinsip Kerja Sistem

Prinsip kerja sistem adalah sebagai berikut:

1. *Power Supply Unit* (PSU) diberi catu 220 VAC.
2. *Power Supply Unit* (PSU) mengkonversi tegangan dari 220 VAC menjadi 12 VDC sebagai suplai rangkaian *driver* motor dan motor pompa air DC.
3. Catu daya Arduino Mega 2560 diambil dari komputer dengan menghubungkan port komunikasi Antara Arduino dan komputer menggunakan perantara kabel USB dengan kecepatan transfer data sebesar 9600 *bit per second* (bps).
4. Sistem diberi catu daya. Catu daya sebesar 5V dari mikrokontroler digunakan untuk mencatu rangkaian penguat sensor arus, sensor tegangan, sensor ping ultrasonik dan mikrokontroler Arduino Mega 2560.
5. Sinyal kontrol dari Arduino Mega 2560 masuk ke *driver* motor EMS 5A H-Bridge. *Driver* motor berfungsi untuk menguatkan sinyal yang dihasilkan mikrokontroler Arduino Mega 2560 dari 0-5 V menjadi 0-12 V.
6. Sinyal keluaran rangkaian penguat sebagai masukan Arduino Mega 2560 yang kemudian diproses menggunakan controller PID.
7. Keluaran Arduino Mega 2560 berupa sinyal pulse width modulation (PWM) *Pulse Width Modulation* diberikan ke *driver* motor.
8. Menggunakan aktuator berupa motor pompa air DC 12V yang akan mengalirkan air melalui pipa dengan diameter selang 0.5 inchi menuju tandon penampungan air diatas gedung bertingkat dengan debit 4.56 liter/menit.
9. Mencari fungsi alih *Plant* dengan menggunakan metode *Ziegler-nichols* 1 dan di dapat grafik kurva s untuk mencari nilai L dan T.
10. Mengimplementasikan hasil desain perancangan pada sistem.

3.6 Pengujian Driver Motor EMS H-Bridge 5A

a. Tujuan

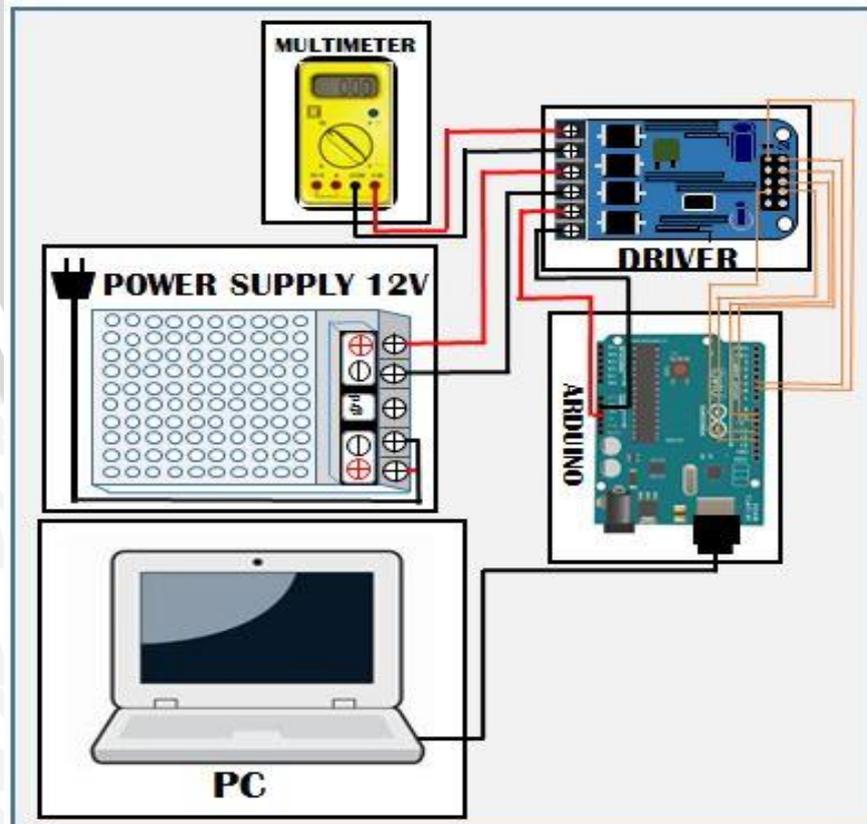
Pengujian ini bertujuan untuk Mengetahui kinerja dan respon rangkaian modul *driver* motor EMS H-Bridge 5A dengan membandingkan *output* tegangan efektif *driver* dengan masukan *dutycycle* sinyal PWM yang diberikan oleh Arduino Mega 2560.

b. Peralatan yang digunakan

1. Seperangkat Komputer.
2. Power Supply.
3. *Driver* EMS H-Bridge 5A.
4. Arduino Mega 2560.
5. Multimeter.
6. Kabel Penghubung.

c. Prosedur pengujian

1. Tulis listing program pada software Arduino untuk mengubah *duty cycle*.
2. Hubungkan *output* tegangan *Power Supply Unit* pada *input* tegangan referensi *driver* motor EMS H-Bridge 5A seperti pada gambar 3.12.
3. Hubungkan *input* tegangan *driver* motor EMS H-Bridge 5A dengan pin *output* PWM di Arduino Mega 2560 seperti pada gambar 3.12.
4. Hubungkan *output* tegangan *driver* motor EMS H-Bridge 5A dengan multimeter seperti pada gambar 3.12.
5. Atur *duty cycle* sinyal PWM pada Arduino Mega dengan nilai 0%-100%.
6. Amati dan catat hasil pembacaan multimeter di setiap kenaikan 5%.



Gambar 3.12 Pengujian *driver* motor EMS H-Bridge 5A (Perancangan)

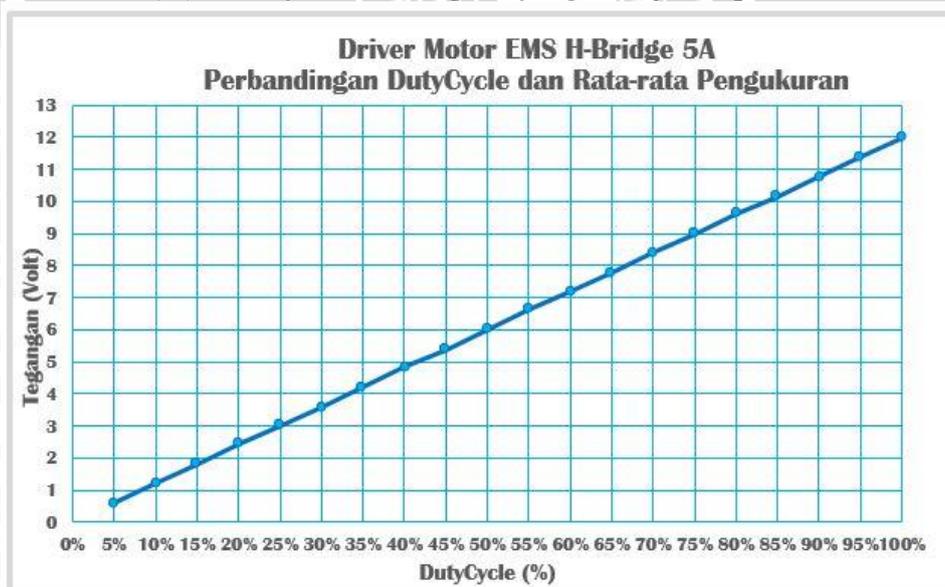
d. Hasil Pengujian

Data pengujian *Driver Motor EMS H-Bridge 5A*, seperti yang ditunjukkan pada Tabel 3.1.

NO	DutyCycle	PWM (Arduino)	Pengukuran 1 (Volt)	Pengukuran 2 (Volt)	Pengukuran 3 (Volt)	Rata-rata (Volt)
1	0%	0	0	0	0	0
2	5%	12.75	0.59	0.59	0.6	0.593
3	10%	25.5	1.2	1.21	1.2	1.203
4	15%	38.25	1.81	1.81	1.82	1.813
5	20%	51	2.42	2.43	2.42	2.423
6	25%	63.75	2.99	2.98	2.99	2.986
7	30%	76.5	3.6	3.6	3.6	3.6
8	35%	89.25	4.21	4.2	4.21	4.206
9	40%	102	4.82	4.81	4.82	4.816
10	45%	114.75	5.38	5.39	5.38	5.383
11	50%	127.5	5.99	5.99	5.99	5.99
12	55%	140.25	6.6	6.6	6.6	6.6
13	60%	153	7.21	7.22	7.21	7.213
14	65%	165.75	7.78	7.77	7.77	7.773
15	70%	178.5	8.38	8.38	8.38	8.38
16	75%	191.25	8.99	9	8.99	8.993
17	80%	204	9.6	9.59	9.6	9.596
18	85%	216.75	10.16	10.15	10.16	10.156
19	90%	229.5	10.77	10.75	10.77	10.763
20	95%	242.25	11.37	11.38	11.38	11.376
21	100%	255	11.98	11.98	11.99	11.983

Tabel 3.1 Hasil Pengujian *driver motor EMS H-Bridge 5A*

Berdasarkan tabel 3.1, akan didapatkan kurva *output* tegangan *driver* (Volt) terhadap masukan *duty cycle* sinyal PWM, seperti yang ditunjukkan pada Gambar 3.13.



Gambar 3.13 Grafik perubahan tegangan *output driver* motor EMS H-Bridge 5A terhadap *input duty cycle*

Sesuai dengan dengan hasil grafik pada Gambar 3.14 dapat disimpulkan bahwa modul *driver* EMS H-Bridge 5A memiliki *gain* sebesar:

$$m = \frac{V_{out_{13}} - V_{out_9}}{VDutyCycle_{13} - VDutyCycle_9} = \frac{7.213 - 4.816}{60 - 40} = 0.11$$

3.7 Pengujian *Driver* Motor EMS H-Bridge 5A dengan Beban Motor Pompa Air DC

a. Tujuan

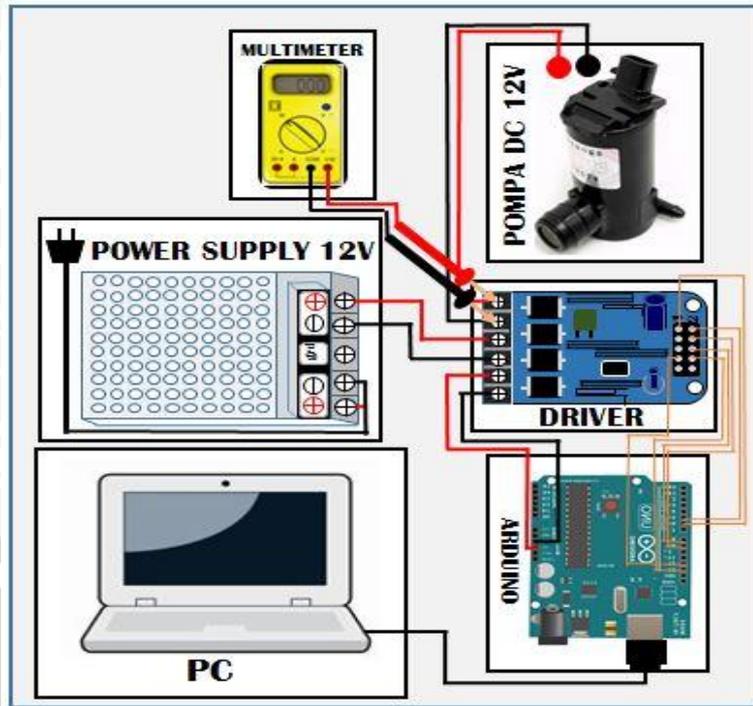
Mengetahui *range* atau karakteristik kerja dari motor pompa DC 12 V dengan *input* berupa *duty cycle* di setiap kenaikan 5%.

b. Peralatan yang digunakan

1. Seperangkat Komputer
2. Motor Pompa DC 12V.
3. *Power Supply Unit*.
4. *Driver* EMS H-Bridge 5A.
5. Arduino Mega 2560.
6. Multimeter.
7. Kabel Penghubung.

c. Prosedur Pengujian

1. Tulis listing program pada software Arduino untuk mengubah *duty cycle*.
2. Hubungkan *output* tegangan *Power Supply Unit* pada *input* tegangan referensi *driver* motor EMS H-Bridge 5A seperti pada gambar 3.14.
3. Hubungkan *input* tegangan *driver* motor EMS H-Bridge 5A dengan pin *output* PWM di Arduino Mega 2560 seperti pada gambar 3.15.
4. Hubungkan *output* tegangan *driver* motor EMS H-Bridge 5A dengan motor pompa DC 12V.
5. Mencatat hasil keluaran motor pompa DC 12V dengan multimeter pada *output* tegangan *driver* motor EMS H-Bridge 5A.
6. Atur *duty cycle* sinyal PWM pada Arduino Mega dengan nilai 0%-100%.
7. Amati dan catat hasil pembacaan multimeter di setiap kenaikan 5%.



Gambar 3.14 Driver EMS H-Bridge 5A dengan menggunakan beban motor pompa DC 12V (Perancangan)

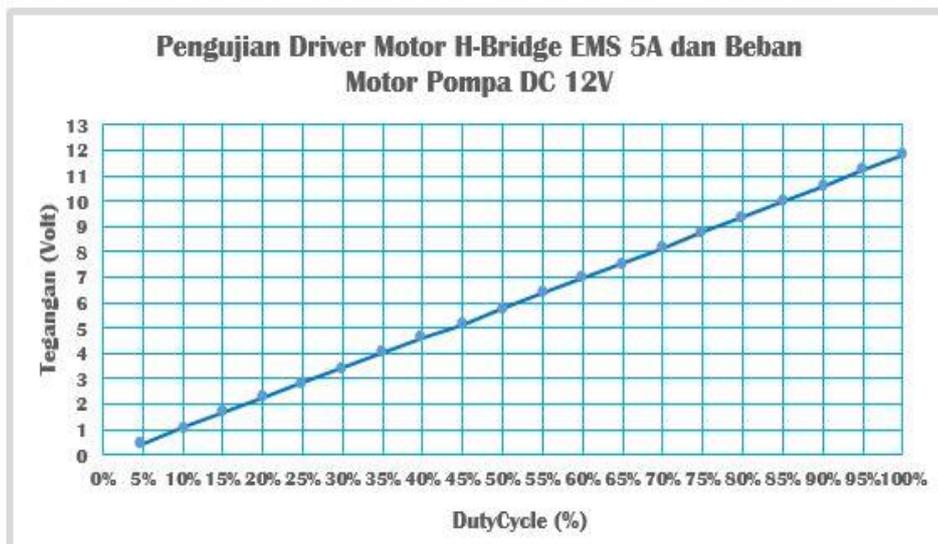
d. Hasil Pengujian

Data hasil pengujian motor pompa DC, seperti yang ditunjukkan pada Tabel 3.2.

NO	DutyCycle	PWM (Arduino)	Pengukuran 1 (Volt)	Pengukuran 2 (Volt)	Pengukuran 3 (Volt)	Rata-rata (Volt)
1	0%	0	0	0	0	0
2	5%	12.75	0.38	0.41	0.41	0.4
3	10%	25.5	1.12	1.11	1.11	1.113
4	15%	38.25	1.7	1.71	1.7	1.703
5	20%	51	2.28	2.29	2.29	2.286
6	25%	63.75	2.82	2.84	2.83	2.83
7	30%	76.5	3.41	3.43	3.42	3.42
8	35%	89.25	4.01	4.02	4.02	4.016
9	40%	102	4.6	4.62	4.61	4.61
10	45%	114.75	5.15	5.17	5.17	5.163
11	50%	127.5	5.76	5.78	5.78	5.773
12	55%	140.25	6.37	6.38	6.38	6.376
13	60%	153	6.99	7	6.99	6.993
14	65%	165.75	7.55	7.56	7.56	7.556
15	70%	178.5	8.17	8.17	8.17	8.17
16	75%	191.25	8.79	8.79	8.79	8.79
17	80%	204	9.4	9.41	9.41	9.406
18	85%	216.75	9.97	9.96	9.98	9.97
19	90%	229.5	10.58	10.59	10.59	10.586
20	95%	242.25	11.19	11.2	11.2	11.196
21	100%	255	11.8	11.79	11.8	11.796

Tabel 3.2 Hasil pengujian driver motor EMS H-BRIDGE 5A dengan menggunakan beban motor pompa DC 12V pada perubahan tegangan rata-rata (Volt) terhadap *duty*cycle (%).

Berdasarkan tabel 3.2, akan didapatkan kurva *output* tegangan *driver* (Volt) terhadap masukan *dutycycle* sinyal PWM, seperti yang ditunjukkan pada Gambar 3.15.



Gambar 3.15 Grafik perubahan tegangan rata-rata *output driver* EMS H-BRIDGE 5A terhadap *dutycycle* dengan menggunakan beban motor pompa DC 12V

Sesuai dengan dengan hasil grafik pada Gambar 3.15 dapat disimpulkan bahwa modul *driver* EMS H-BRIDGE 5A dengan menggunakan beban motor pompa DC 12V memiliki *gain* sebesar:

$$m = \frac{V_{out_{13}} - V_{out_9}}{VDutycycle_{13} - VDutycycle_9} = \frac{6.993 - 4.61}{60 - 40} = 0.11$$

3.8 Pengujian Sensor Tegangan

a. Tujuan

Mengetahui Karakteristik, kinerja sensor tegangan dengan membandingkan *input* sebesar 0 – 12V yang dibaca oleh sensor tegangan dan *output* sebesar 0-5V yang diberikan oleh Arduino Mega 2560.

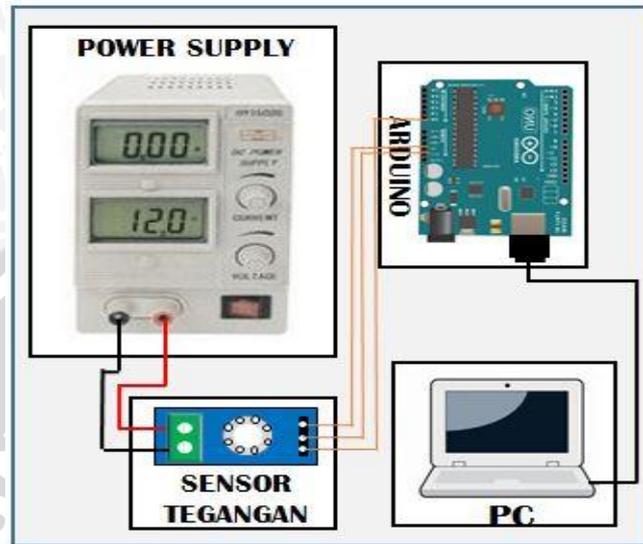
b. Peralatan yang digunakan.

1. Seperangkat Komputer.
2. *Power Supply Unit*
3. Arduino Mega 2560.
4. Sensor Tegangan.
5. Kabel Penghubung.

c. Prosedur Pengujian

1. Membuat listing program untuk sensor tegangan di software arduino.
2. Atur *input* pada sensor tegangan di software Arduino dengan 0-12V.
3. Hubungkan *output* tegangan *Power Supply Unit* pada *input* sensor tegangan seperti pada gambar 3.16

4. Hubungkan *input* sensor tegangan dengan pin di Arduino Mega 2560 seperti pada gambar 3.16.
5. Mencatat hasil *output* sensor tegangan
6. Amati dan catat hasil pembacaan sensor tegangan di setiap kenaikan tegangan 0.5 V.



Gambar 3.16 Hasil Pengujian Sensor Tegangan (Perancangan)

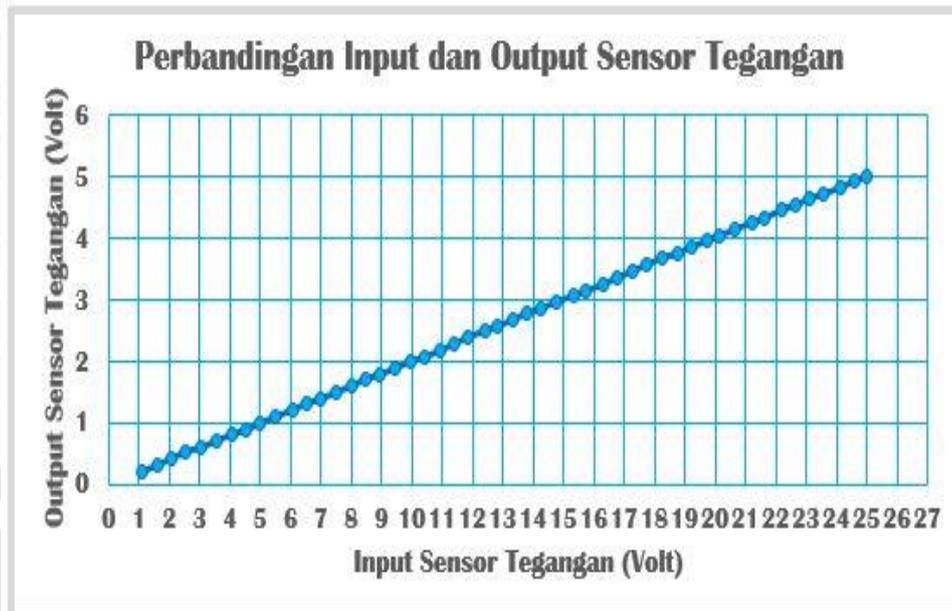
d. Hasil pengujian

Data hasil pengujian motor pompa DC, seperti yang ditunjukkan pada Tabel 3.3.

NO	Masukan Sensor (Volt)	Voltmeter (Volt)	Sensor Tegangan (Volt)	Keluaran Sensor (Volt)
1	0	0	0	0
2	0.5	1.12	1.08	0.22
3	1	1.63	1.59	0.32
4	1.5	2.1	2.05	0.41
5	2	2.59	2.54	0.51
6	2.5	3.07	3.03	0.61
7	3	3.56	3.52	0.7
8	3.5	4.12	4.08	0.82
9	4	4.54	4.5	0.9
10	4.5	5.03	4.99	1
11	5	5.54	5.5	1.1
12	5.5	6.07	6.04	1.21
13	6	6.54	6.5	1.3
14	6.5	7.01	6.96	1.4
15	7	7.54	7.5	1.5
16	7.5	8.01	7.99	1.6
17	8	8.49	8.46	1.69
18	8.5	8.96	8.92	1.78
19	9	9.47	9.43	1.89
20	9.5	9.96	9.95	1.99
21	10	10.42	10.39	2.08
22	10.5	10.93	10.9	2.18
23	11	11.42	11.41	2.28
24	11.5	11.88	11.85	2.37
25	12	12.4	12.39	2.48

Tabel 3.3 Hasil pengujian sensor tegangan

Berdasarkan tabel 3.3 akan didapatkan kurva *output* tegangan pada arduino sebesar 0-5V terhadap *input* tegangan pada sensor sebesar 0-25V, seperti yang ditunjukkan pada Gambar 3.17



Gambar 3.17 Grafik perubahan tegangan *output* pada sensor tegangan

Sesuai dengan dengan hasil grafik pada Gambar 3.20 dapat disimpulkan bahwa modul sensor tegangan memiliki *gain* sebesar:

$$m = \frac{V_{out_{13}} - V_{out_9}}{V_{in_{13}} - V_{in_9}} = \frac{1.3 - 0.9}{6.5 - 4.5} = 0.2$$

3.9 Pengujian Sensor Arus

a. Tujuan

Untuk mengetahui apakah sensor arus yang dirancang dapat bekerja sesuai dengan yang diharapkan dengan cara membandingkan *input* berupa arus saat diberi beban berupa motor pompa air DC 12V dengan *output* sensor arus dihubungkan dengan rangkaian penguat agar dapat dibaca oleh mikrokontroler Arduino Mega 2560 dengan rentang 0-5V.

b. Peralatan yang digunakan

1. Seperangkat Komputer.
2. *Power Supply Unit*.
3. Arduino Mega 2560.
4. Sensor Arus ACS 712 5A.
5. Motor Pompa Air DC 12V.
6. Kabel Penghubung.

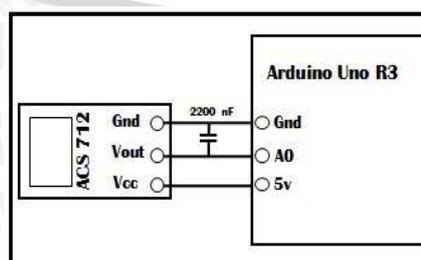
c. Prosedur Pengujian

1. Membuat *listing* program untuk sensor arus di software arduino.
2. Atur *input* pada sensor arus di software Arduino.
3. Hubungkan *output* tegangan *Power Supply Unit* pada *input* sensor arus seperti pada gambar 3.18.
4. Pemasangan sensor arus harus di seri dengan beban motor pompa air DC 12V agar *input* pada saat pengujian dapat terbaca berupa arus yang mengalir pada beban.
5. Hubungkan *input* sensor arus dengan pin di Arduino Mega 2560 seperti pada gambar 3.18.
6. *Output* pada sensor arus sebesar -5-5A yang berupa arus harus dikonversikan menjadi tegangan pada Arduino Mega 2560 sebesar 0-5V.
7. Mencatat hasil *output* sensor arus berupa tegangan 0-5V.
8. Amati dan catat hasil pembacaan sensor arus di setiap kenaikan.



Gambar 3.18 Hasil Pengujian Sensor arus (Perancangan)

perbedaan antara hasil hardware modif dan standart tidak terlalu besar, tetapi pada konfigurasi hardware modif dengan menggunakan kapasitor 2200nF dapat memblock nilai hingga di sekitar 0.04-0.05A.



Gambar 3.19 Skema Modif Sensor Arus

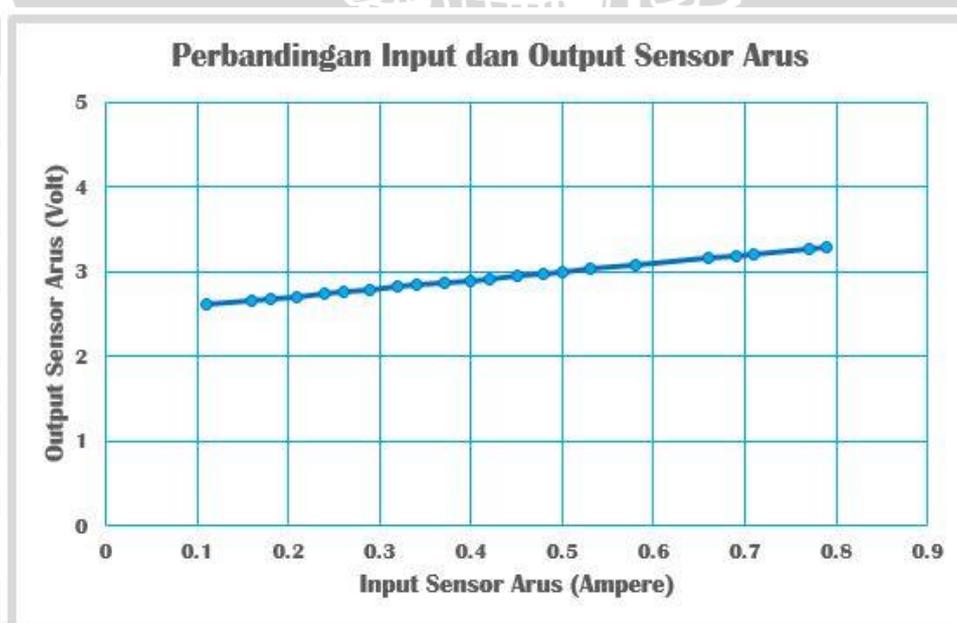
d. Hasil Pengujian

Data hasil pengujian sensor tegangan yang ditunjukkan pada Tabel 3.4.

NO	Ammeter (Ampere)	Sensor Arus (Ampere)	Keluaran Sensor (Volt)
1	0	0	0
2	0.13	0.11	2.61
3	0.16	0.16	2.66
4	0.21	0.18	2.68
5	0.24	0.21	2.71
6	0.32	0.24	2.74
7	0.33	0.26	2.76
8	0.34	0.29	2.79
9	0.38	0.32	2.82
10	0.43	0.34	2.84
11	0.45	0.37	2.87
12	0.52	0.4	2.9
13	0.54	0.42	2.92
14	0.59	0.45	2.95
15	0.63	0.48	2.98
16	0.66	0.5	3
17	0.74	0.53	3.03
18	0.81	0.58	3.08
19	0.9	0.66	3.16
20	0.92	0.69	3.19
21	0.98	0.71	3.21
22	1.08	0.77	3.27
23	1.12	0.79	3.29

Tabel 3.4 Hasil pengujian sensor arus

Pada Gambar grafik 3.21 menunjukkan perbandingan *input* berupa arus terhadap *output* berupa tegangan sebesar 0-5 V dengan mengubah nilai *input* menggunakan *Power Supply Unit* saat di beri beban berupa motor pompa air DC 12 V.



Gambar 3.21 Grafik perubahan tegangan *output* pada sensor arus.

Kesimpulan dari pengujian diatas didapat *gain* atau fungsi alih sensor arus sebesar:

$$m = \frac{V_{out_{13}} - V_{out_7}}{V_{in_{13}} - V_{in_7}} = \frac{2.92 - 2.76}{0.42 - 0.26} = 0.625$$

3.10 Pengujian Sensor Ping Ultrasonik

a. Tujuan

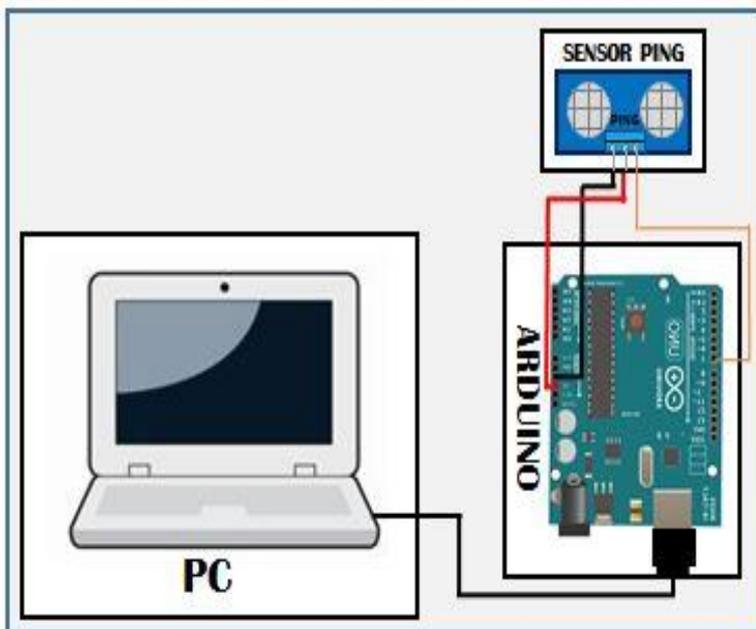
Mengetahui kemampuan sensor jarak ping ultrasonik dalam membaca perubahan jarak *input* sebesar 0-300 centimeters menjadi *output* berupa tegangan yang dibaca oleh Arduino Mega 2560 sebesar 0-5 V.

b. Peralatan yang digunakan

1. Seperangkat Komputer.
2. Power supply unit.
2. Arduino Mega 2560.
3. Sensor Ping Ultrasonik.
4. Kabel penghubung.

c. Prosedur Pengujian

1. Membuat listing program untuk sensor jarak ping ultrasonik di software arduino.
2. Atur *input* pada sensor level ping ultrasonik di software Arduino berupa jarak centimeters.
3. Hubungkan *input* sensor jarak ping ultrasonik dengan pin di Arduino Mega 2560 seperti pada gambar 3.21.
4. Bandingkan jarak sebenarnya yang terukur pada penggaris dengan jarak *input* yang terbaca oleh sensor jarak ping ultrasonik.
5. *Output* pada sensor jarak ping ultrasonik sebesar 0-300 centimeters yang dikonversikan menjadi tegangan pada Arduino Mega 2560 sebesar 0-5 V.
6. Mencatat hasil *output* sensor arus berupa tegangan 0-5 V.
7. Amati dan catat hasil pembacaan sensor jarak ping ultrasonik di setiap kenaikan 10 milimeters.



Gambar 3.21 Hasil Pengujian Sensor jarak ping ultrasonik (Perancangan)

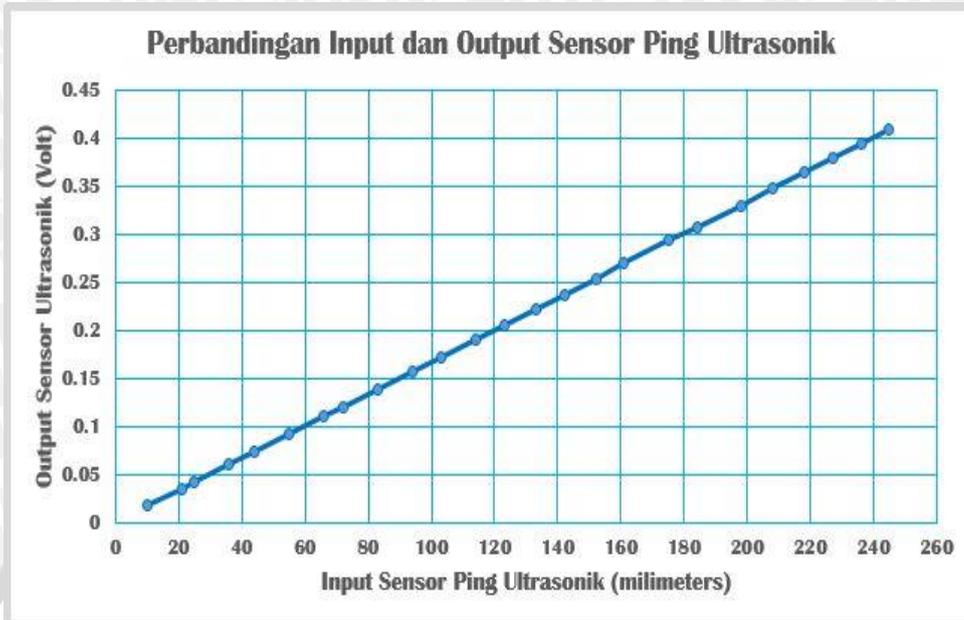
d. Hasil Pengujian

Data hasil pengujian sensor tegangan yang ditunjukkan pada Tabel 3.5.

NO	Jarak Sebenarnya (mm)	Input Sensor Ping (mm)	Output Sensor Ping (Volt)
1	0	0	0
2	10	10	0.017
3	20	21	0.035
4	30	25	0.042
5	40	36	0.06
6	50	44	0.073
7	60	55	0.092
8	70	66	0.11
9	80	72	0.12
10	90	83	0.138
11	100	94	0.156
12	110	103	0.172
13	120	114	0.19
14	130	123	0.205
15	140	133	0.222
16	150	142	0.237
17	160	152	0.253
18	170	161	0.269
19	180	175	0.294
20	190	184	0.307
21	200	198	0.33
22	210	208	0.347
23	220	218	0.364
24	230	227	0.379
25	240	236	0.394
26	250	245	0.409

Tabel 3.5 Hasil pengujian sensor jarak ping ultrasonik.

Pada Gambar grafik 3.22 menunjukkan perbandingan *input* berupa ketinggian terhadap *output* berupa tegangan sebesar 0-5 V.



Gambar 3.22 Grafik perubahan tegangan *output* pada sensor jarak ping ultrasonik

Kesimpulan dari pengujian diatas didapat *gain* atau fungsi alih sensor ping sebesar:

$$m = \frac{V_{out_{13}} - V_{out_{11}}}{V_{in_{13}} - V_{in_{11}}} = \frac{0.19 - 0.156}{114 - 94} = 1.7 \times 10^{-3}$$

3.11 Pengujian Motor Pompa Air DC 12V

a. Tujuan

Mengetahui karakteristik pompa air DC 12V, dengan memberikan nilai *input* berupa *dutycycle* sebesar 5%-100% menjadi *output* berupa debit (l/s) dengan membandingkan volume yang terukur sebesat 1000 ml dengan waktu (sekon) yang dibutuhkan selama volume terisi air sesuai dengan yang sudah ditentukan.

b. Peralatan yang digunakan

1. Seperangkat Komputer.
2. *Power Supply Unit*.
3. Arduino Mega 2560.
4. Bejana ukur (liter).
5. Kabel penghubung.
6. *Driver EMS H-Bridge 5A*.
7. Motor pompa air DC 12V.
8. Kabel Penghubung

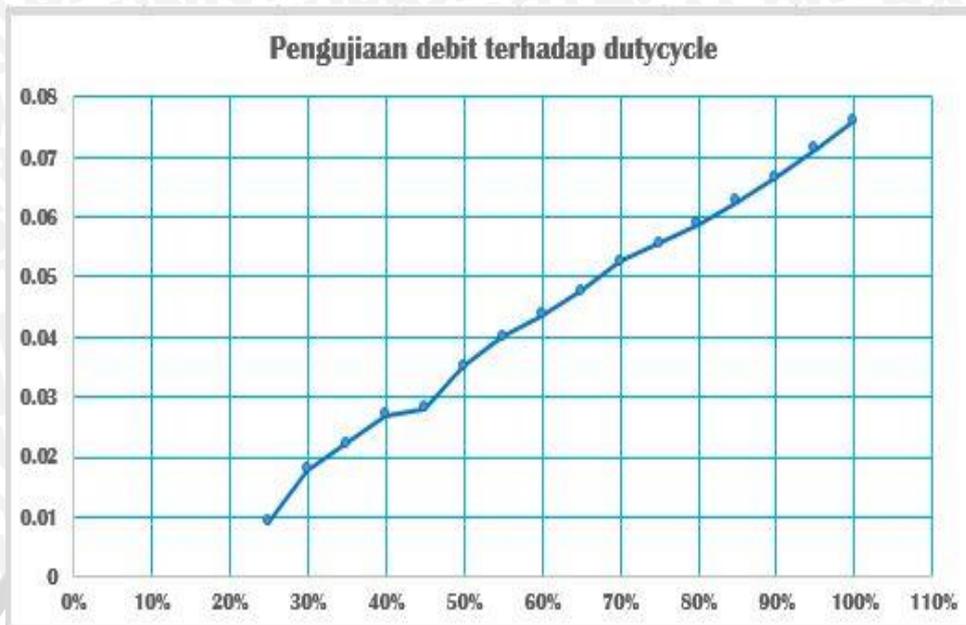
c. Prosedur Pengujian

1. Membuat listing program dengan mengatur *dutycycle* 5%-100%.
2. Atur *input* pada *driver* motor di software Arduino berupa *dutycycle*.
3. *Driver* motor mendapatkan catu dari power supply unit sebesar 0-12V.
4. Hubungkan *driver* motor dengan pin di Arduino mega 2560.
5. Hubungkan *output driver* dengan pompa air DC 12V.
6. Bandingkan volume yang terukur dengan waktu pada saat pengisian air ke dalam bejana.
7. Maka di dapatkan debit air sesuai dengan perhitungan.
8. Mencatat hasil *output* berupa debit air (l/s).
9. Amati dan catat karakteristik pompa air DC 12V disetiap kenaikan *dutycycle* 5%.

NO	Duty cycle	Vin Driver (Teori)	Vout Drive (Teori)	Vin Driver (Data)	Vout Drive (Data)	PWM	Volume (ml)	Waktu (Sekon)	Percobaan Debit (l/s)
1	5%	0.25	0.6	0.25	0.4	12.75	1000	0	0
2	10%	0.5	1.2	0.5	1.08	25.5	1000	0	0
3	15%	0.75	1.8	0.75	1.68	38.25	1000	0	0
4	20%	1	2.4	1.01	2.28	51	1000	0	0
5	25%	1.25	3	1.24	2.81	63.75	1000	109d	0.00917
6	30%	1.5	3.6	1.49	3.38	76.5	1000	56d	0.0178
7	35%	1.75	4.2	1.74	3.92	89.25	1000	45d	0.0222
8	40%	2	4.8	1.99	4.47	102	1000	36d	0.027
9	45%	2.25	5.4	2.22	4.94	114.75	1000	35d	0.028
10	50%	2.5	6	2.47	5.51	127.5	1000	28d	0.035
11	55%	2.75	6.6	2.73	6.06	140.25	1000	25d	0.04
12	60%	3	7.2	2.98	6.61	153	1000	23d	0.0437
13	65%	3.25	7.8	3.22	7.13	165.75	1000	21d	0.0476
14	70%	3.5	8.4	3.47	7.68	178.5	1000	19d	0.0526
15	75%	3.75	9	3.72	8.23	191.25	1000	18d	0.0555
16	80%	4	9.6	3.95	8.78	204	1000	17d	0.05882
17	85%	4.25	10.2	4.2	9.29	216.75	1000	16d	0.0625
18	90%	4.5	10.8	4.46	9.72	229.5	1000	15d	0.0666
19	95%	4.75	11.4	4.7	10.29	242.25	1000	14d	0.0714
20	100%	5	12	4.96	10.97	255	1000	13d	0.076

Tabel 3.6 Hasil pengujian debit air

Pada gambar grafik 3.23 menunjukkan perbandingan *output* berupa debit (l/s) terhadap *input* berupa *dutycycle*.



Gambar 3.23 Grafik perubahan *output* terhadap *dutycycle*

Kesimpulan dari pengujian diatas didapat *gain* atau fungsi alih motor DC 12V sebesar:

$$m = \frac{V_{out_{18}} - V_{out_{14}}}{V_{in_{18}} - V_{in_{14}}} = \frac{90 - 70}{0.0666 - 0.0526} = 1.428$$

3.12 Mencari *Steady State Gain* pada *Plant*

a. Tujuan

Steady state gain pada *Plant* untuk menemukan fungsi alih *Plant* dan mendapatkan karakteristik statis *gain* keadaan mantap dengan cara *Plant* di rancang secara *open loop* dan diberikan unit step masukan *dutycycle* 0%–100% dan di lihat bagaimana respon keadaan mantapnya.

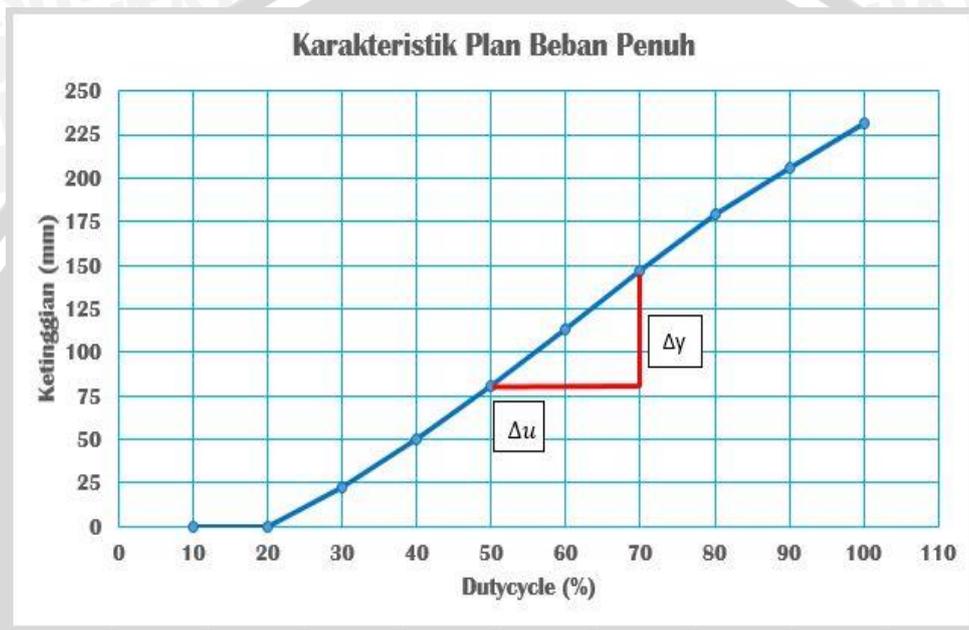
b. Peralatan yang digunakan

1. Seperangkat komputer.
2. *Power supply unit*.
3. Arduino Mega 2560.
4. *Driver* EMS H-Bridge 5A.
5. Motor pompa air DC 12V.
6. Kabel Penghubung.

c. Prosedur Pengujian

1. Membuat listing program dengan mengatur *dutycycle* 0%-100%.

2. Atur *input* pada *driver* motor di software Arduino berupa *dutycycle*.
3. *Driver* motor mendapatkan catu dari *Power Supply Unit* 0-12V.
4. Hubungkan *driver* motor dengan pin di Arduino Mega 2560.
5. Hubungkan *output driver* dengan pompa air DC 12V.
6. Buka beban maksimal pada plan berupa tandon penyimpanan air dengan memberikan masukan berupa *dutycycle* sebesar 0%–100% untuk mengetahui respon *steadystate* dalam keadaan mantap, seperti yang ditunjukkan pada gambar 3.24.



Gambar 3.24 Grafik Karakteristik statis *gain* keadaan mantap

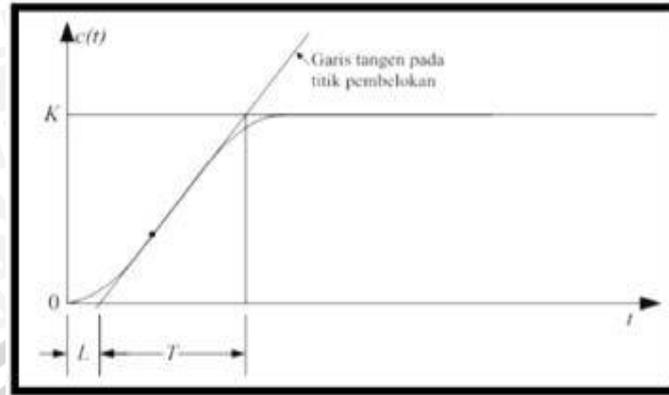
Dari grafik diatas, dapat disimpulkan bahwa nilai *steady state gain* keadaan mantap sebesar:

$$k = \frac{\Delta y}{\Delta u} = \frac{146.8 - 81}{70 - 50} = 3.29$$

Untuk memenuhi tujuan performansi *loop* yang diinginkan, maka perlu ditambahkan kontroler pada sistem tersebut. Kontroler yang dipilih adalah Proporsional Integral Derivatif (PID). Kontroler PID dipilih karena karakteristik *Plant* tersebut membutuhkan kontrol yang cepat dan meminimalkan *overshoot* untuk dapat menghasilkan level ketinggian air agar sesuai dengan *setpoint* sesuai yang diinginkan.

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).

Pada pengerjaan skripsi ini digunakanlah metode *Ziegler-nichols* yang pertama sehingga mendapatkan sinyal kurve S seperti pada Gambar 3.25.



Gambar 3.25 Kurva Respon yang berbentuk S
(Sumber: Katsuhiko Ogata, 1997)

Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda L dan konstanta waktu T . Waktu tunda dan konstanta waktu ditentukan dengan menggambar sebuah garis tangen dengan sumbu waktu t dan sumbu $c(t) = K$.

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

1. Menarik garis *tangent* pada titik infleksi pada grafik karakteristik *Plant* seperti pada Gambar 3.25.
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 , T_d sesuai ketentuan pada Gambar 3.7

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

Tabel 3.7 Aturan Metode Ziegler-nichols 1
(Sumber: Katsuhiko Ogata, 1997)

3.13 Perancangan Parameter PID

a. Tujuan

Pada perancangan PID

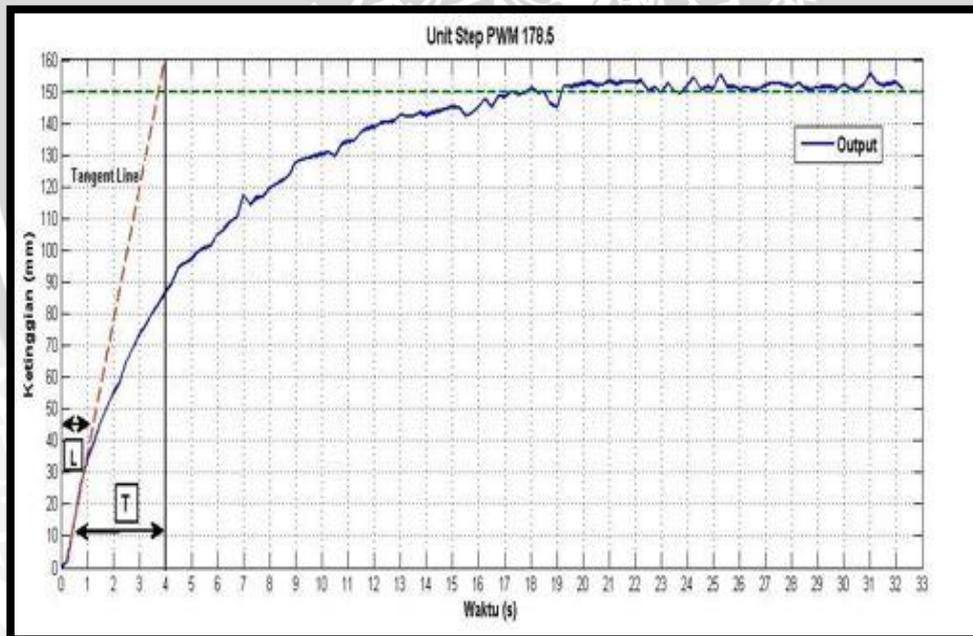
Pengujian keseluruhan sistem dilakukan apakah parameter yang sudah ditentukan dapat diaplikasikan pada alat dan sudah sesuai dengan *setpoint* yang diinginkan.

b. Langkah Pengujian

Adapun langkah pengujian keseluruhan sistem untuk menemukan parameter kontroler PID digunakan metode *ziegler-nichols* 1:

1. Diberi tegangan masukan ke *Plant* untuk mendapatkan nilai L dan T.
2. Memasukkan nilai parameter K_p , K_i dan K_d dari hasil perhitungan.
3. Memasukkan *setpoint* yang diinginkan.
4. Atur pembukaan katub keluaran air sebagai beban dan beban lebih selama 150 detik sebagai gangguan.
5. Mengamati dan menganalisa hasil dari kinerja sistem.

Berikut merupakan kurva S hasil pengujian sistem keseluruhan dengan menggunakan metode *Ziegler-nichols* 1 yang ditunjukkan dalam Gambar 3.26.



Gambar 3.26 Grafik *Open loop Respon* PWM 178.5

Dari grafik *output Plant* pada Gambar 4.4 didapatkan nilai:

$$\Delta u = \text{PWM } 178.5 \quad L = 0.23 \text{ s}$$

$$\Delta y = 150 \text{ mm} \quad T = 3.87 \text{ s}$$

Setelah mendapatkan parameter yang dibutuhkan maka bias didapatkan fungsi alih *plant* seperti yang ditunjukkan dalam persamaan :

$$G(s) = \frac{Ke^{-Ls}}{Ts + 1} = \frac{2.19e^{-0.23s}}{3.87s + 1}$$

Dan diperoleh nilai K_p , K_i , K_d :

$$K_p = 1.2 \times \frac{T}{L} = 1.2 \times \frac{3.87}{0.23} = 20.19$$

$$K_i = \frac{K_p}{T_i} = \frac{20.19}{2 \times L} = \frac{20.19}{0.46} = 43.89$$

$$K_d = K_p \times T_d = 20.19 \times (0.5 \times L) = 20.19 \times 0.115 = 2.32$$

Dari perolehan nilai K_p , K_i , K_d maka didapatkan persamaan transformasi laplace kontroler PID sebagai berikut:

$$C(s) = \left(K_p + \frac{K_i}{s} + K_d s \right) e(s) \dots \dots \dots (3.1)$$

Persamaan diatas belum bisa dimasukkan kedalam mikrokontroler, maka persamaan kontinyu di atas baru diubah kedalam diskrit melalui Transformasi Z. Dalam Transformasi Z dibutuhkan waktu sampling (T_s). Digunakan metode *Backward Difference* sehingga nilai notasi s pada Laplace setara dengan

$$s = \frac{1-z^{-1}}{T_s} \dots \dots \dots (3.2)$$

Bila persamaan (2) disubstitusikan ke dalam persamaan (1) menjadi:

$$\begin{aligned} C(z) &= \left[K_p + \frac{K_i}{\frac{(1-z^{-1})}{(T_s)}} + K_d \frac{(1-z^{-1})}{T_s} \right] E(z) \\ &= \left[K_p + \frac{K_i \times T_s}{(1-z^{-1})} + \frac{K_d}{T_s} (1-z^{-1}) \right] E(z) \end{aligned}$$

Kontroler Proporsional:

$$C_p(z) = K_p E(z)$$

Kontroler Integral:

$$C_i(z) = \left(\frac{K_i T_s}{1-z^{-1}} \right) x E(z)$$

$$C_i(z) - C_i(z)z^{-1} = K_i T_s x E(z)$$

$$C_i(z) = C_i(z)z^{-1} + K_i T_s x E(z)$$

Kontroler Diferensial:

$$C_d(z) = \frac{K_d}{T_s} (z)(1 - z^{-1})x E(z)$$

$$C_d(z) = \frac{K_d}{T_s} [E(z) - E(z)z^{-1}]$$

Dari persamaan diatas kemudian diubah kedalam persamaan beda sehingga didapatkan persamaan dibawah ini:

Kontroler Proporsional : $C_p(k) = K_p x E(k)$

Kontroler Integral : $C_i(k) = C_i(k - 1) + K_i T_s x E(k)$

Kontroler Diferensial : $C_d(k) = \frac{K_d}{T_s} [E(k) - E(k - 1)]$

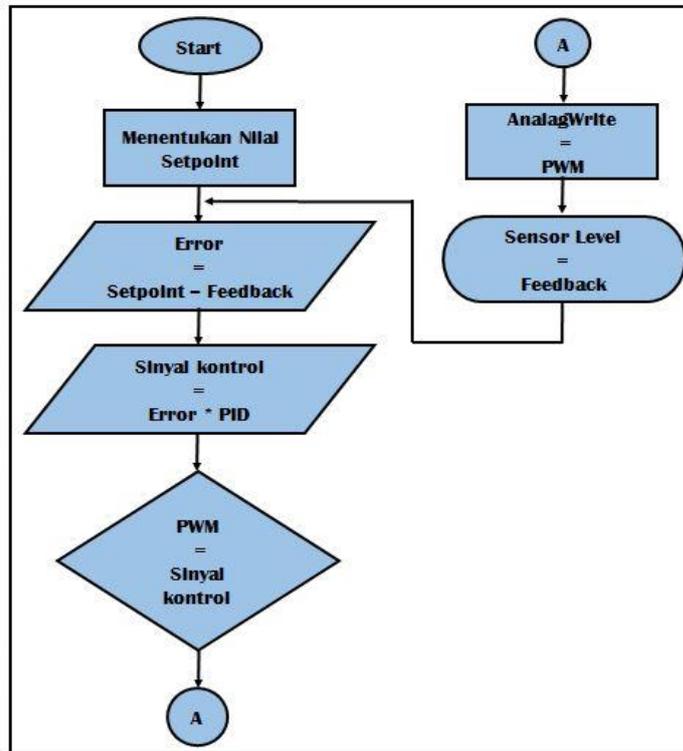
Jika digabungkan menjadi:

$$C(k) = [K_p x E(k)] + [C_i(k - 1) + K_i T_s x E(k)] + \left[\frac{K_d}{T_s} [E(k) - E(k - 1)] \right]$$

Dimana (k-1) adalah kondisi sebelumnya. Persamaan di atas telah berupa persamaan diskrit sehingga dapat dimasukkan ke dalam mikrokontroler.

3.14 Flowchart Menggunakan Kontroler PID

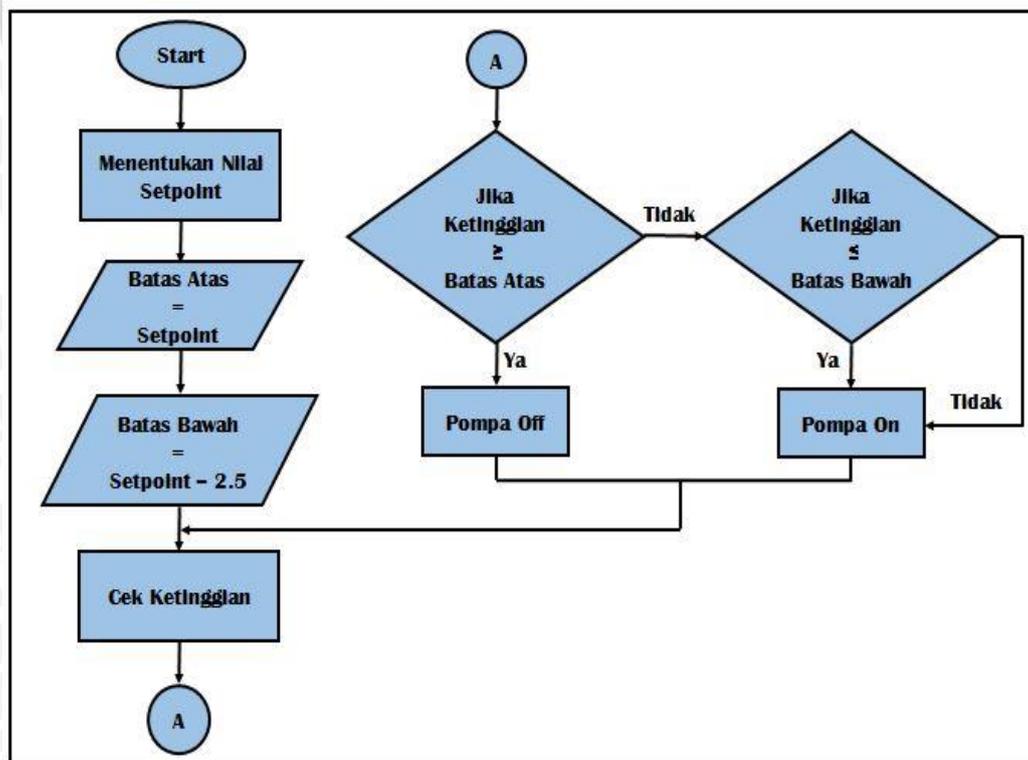
Flowchart menggunakan Kontroler PID ditunjukkan pada Gambar 3.27.



Gambar 3.27 Flowchart Keseluruhan Sistem dengan Kontroler PID

3.11 Flowchart menggunakan Kontroler On-Off

Flowchart menggunakan Kontroler On-Off ditunjukkan pada Gambar 3.28.



Gambar 3.28 Flowchart Keseluruhan Sistem dengan kontroler On-Off

BAB IV

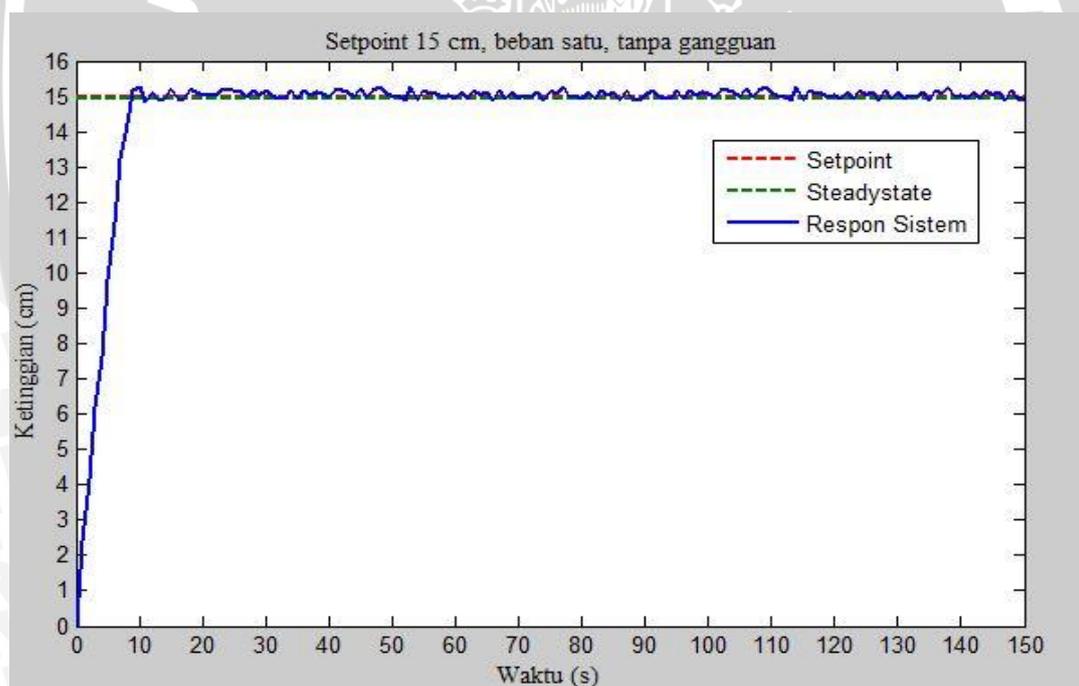
HASIL DAN PEMBAHASAN

Pada bab ini membahas hasil dari pengujian sistem secara keseluruhan baik menggunakan kontroler PID maupun kontroler *on-off*. Pengujian dilakukan untuk mengetahui apakah sistem yang dirancang dapat bekerja sesuai dengan yang diharapkan.

4.1 Pengujian Sistem Dengan Kontroler PID

Pada hasil pengujian menggunakan kontroler PID dengan metode *Ziegler Nichols* 1 didapatkan nilai $K_p=20,19$ $K_i=43,4$ dan $K_d=2,3$ guna mengetahui hasil respon sistem dan penggunaan pemakaian daya elektrik dengan variasi beban maupun variasi beban dengan gangguan.

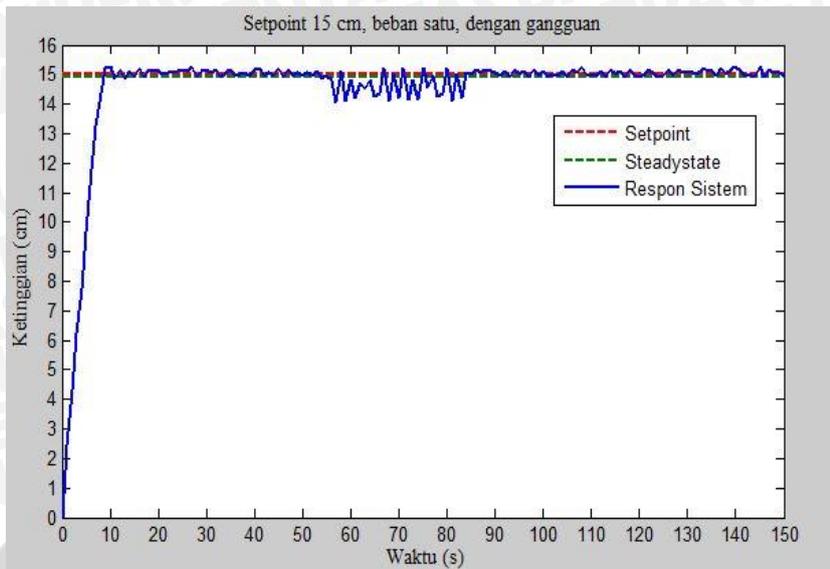
a) Pengujian pada *setpoint* 15 cm, beban satu, tanpa gangguan.



Gambar 4.1 Grafik respon sistem pada *setpoint* 15 cm, beban satu, tanpa gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.1 dapat disimpulkan bahwa pemakaian daya elektrik sebesar 0.861 Watt dengan waktu keadaan mantap (t_s) selama 14 detik, *error steady state* 0,467 %, waktu naik (t_r) selama 9 detik, waktu tunda (t_d) selama 3 detik, waktu puncak (t_p) selama 10 detik dan maksimum *overshoot* (M_p) 1,7333%.

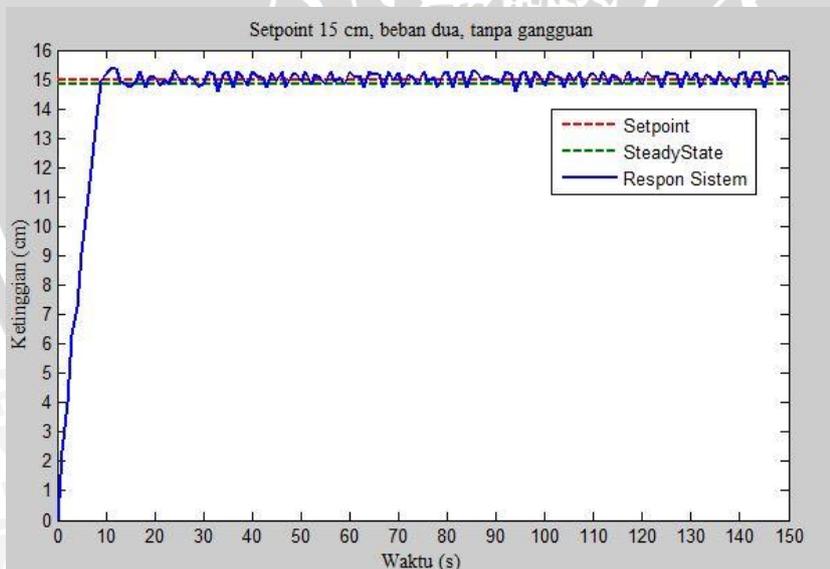
b) Pengujian pada *setpoint* 15 cm, beban satu, dengan gangguan.



Gambar 4.2 Grafik respon sistem pada *setpoint* 15 cm, beban satu, dengan gangguan.

Pada grafik respon sistem seperti yang ditunjukkan dalam Gambar 4.2 dengan adanya gangguan selama 28 detik dapat disimpulkan bahwa pemakaian energi elektrik sebesar 1,35 Watt dengan waktu keadaan mantap (t_s) selama 15 detik, *error steady state* 0,534 % dan maksimum *overshoot* (M_p) 1,866 %.

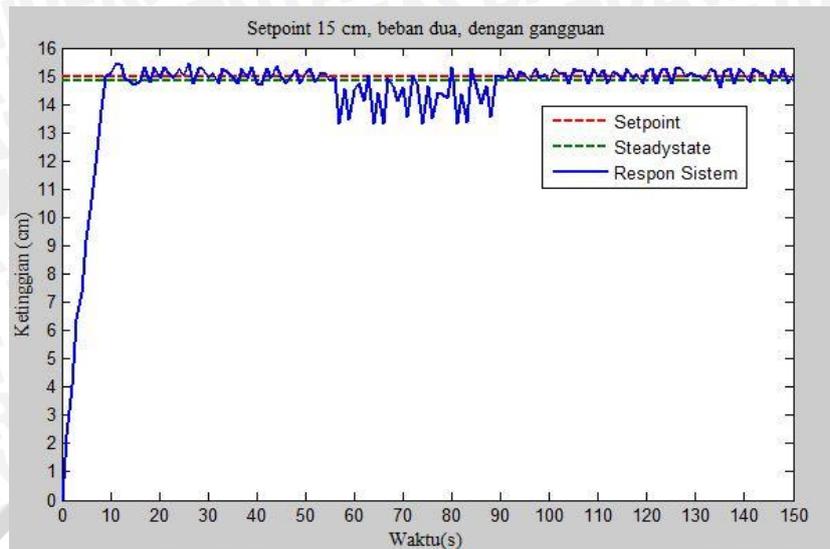
c) Pengujian pada *setpoint* 15 cm, beban dua, tanpa gangguan.



Gambar 4.3 Grafik respon sistem pada *setpoint* 15 cm, beban dua, tanpa gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.3 dapat disimpulkan bahwa pemakaian daya elektrik sebesar 1,365 Watt dengan waktu keadaan mantap (t_s) selama 18 detik, *error steady state* 0,9 %, waktu naik (t_r) selama 10 detik, waktu tunda (t_d) selama 4 detik, waktu puncak (t_p) selama 12 detik dan maksimum *overshoot* (M_p) 2,467 %.

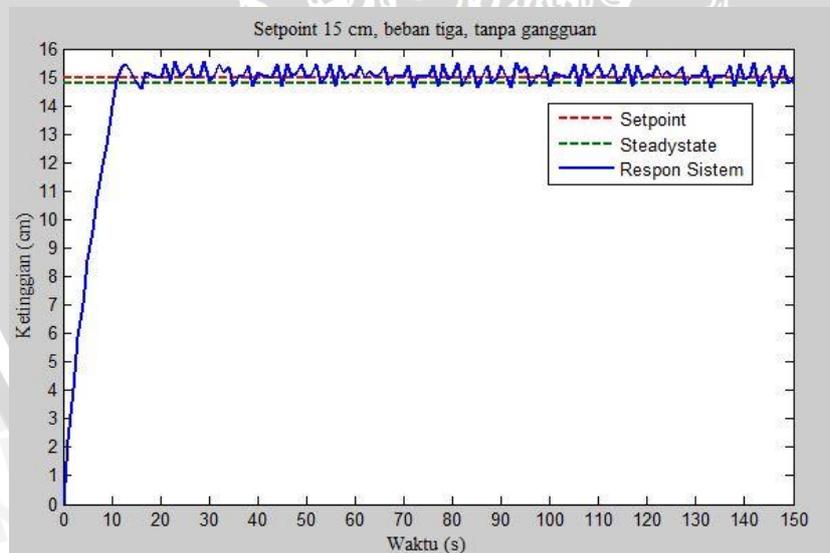
d) Pengujian pada *setpoint* 15 cm, beban dua, dengan gangguan.



Gambar 4.4 Grafik respon sistem pada *setpoint* 15 cm, beban dua, dengan gangguan.

Pada grafik respon sistem seperti yang ditunjukkan dalam Gambar 4.4 dengan adanya gangguan selama 33 detik dapat disimpulkan bahwa pemakaian energi elektrik sebesar 1,927 Watt dengan waktu keadaan mantap (t_s) selama 19 detik, *error steady state* 1 % dan maksimum *overshoot* (M_p) 2,6 %.

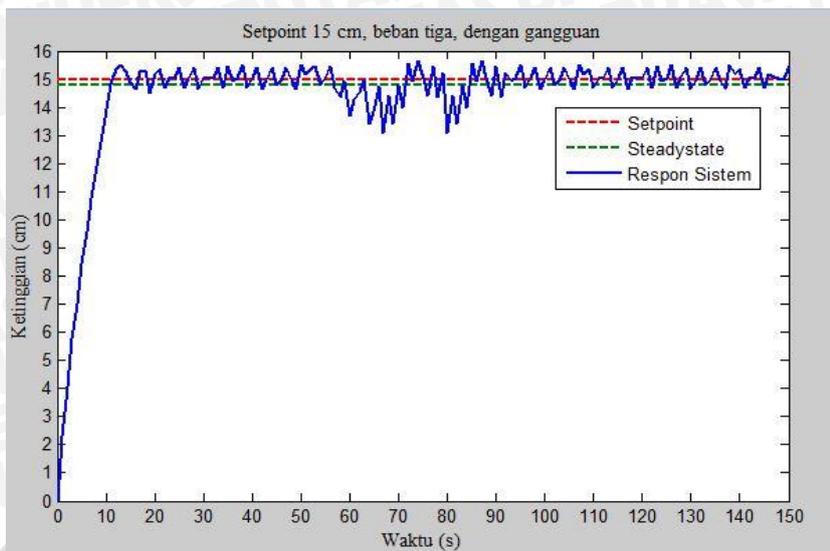
e) Pengujian pada *setpoint* 15 cm, beban tiga, tanpa gangguan.



Gambar 4.5 Grafik respon sistem pada *setpoint* 15 cm, beban tiga, tanpa gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.5 dapat disimpulkan bahwa pemakaian daya elektrik sebesar 2,985 Watt dengan waktu keadaan mantap (t_s) selama 20 detik, *error steady state* 1,167 %, waktu naik (t_r) selama 12 detik, waktu tunda (t_d) selama 5 detik, waktu puncak (t_p) selama 13 detik dan maksimum *overshoot* (M_p) 3.133%.

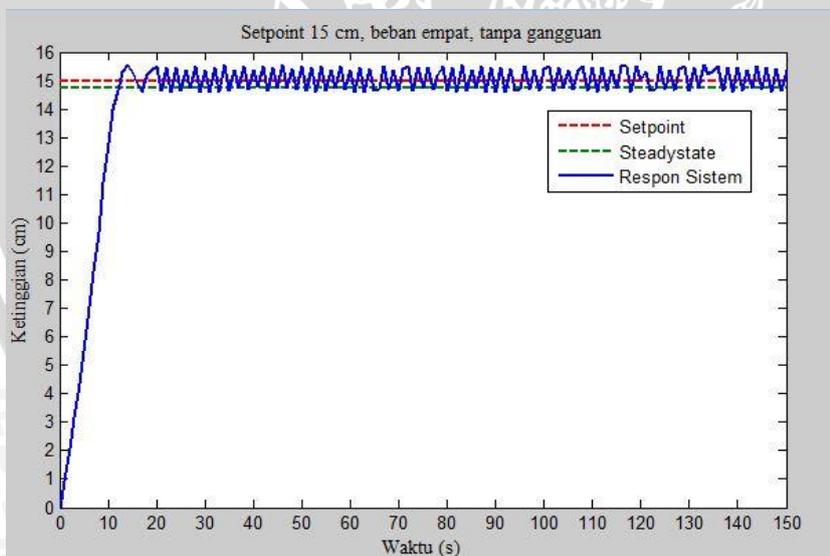
f) Pengujian pada *setpoint* 15 cm, beban tiga, dengan gangguan.



Gambar 4.6 Grafik respon sistem pada *setpoint* 15 cm, beban tiga, dengan gangguan.

Pada grafik respon sistem seperti yang ditunjukkan dalam Gambar 4.6 dengan adanya gangguan selama 30 detik dapat disimpulkan bahwa pemakaian energi elektrik sebesar 3,277 Watt dengan waktu keadaan mantap (t_s) selama 21 detik, *error steady state* 1.3 % dan maksimum *overshoot* (M_p) 3,266 %.

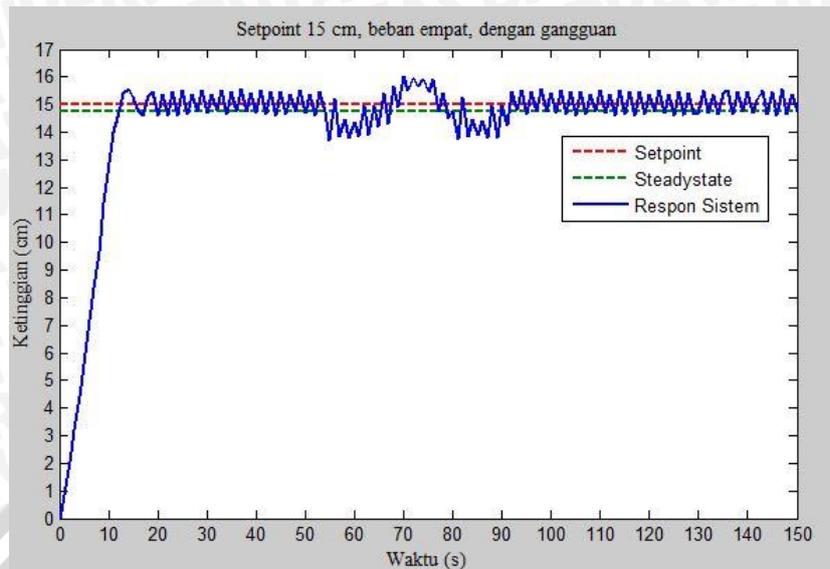
g) Pengujian pada *setpoint* 15 cm, beban empat, tanpa gangguan.



Gambar 4.7 Grafik respon sistem pada *setpoint* 15 cm, beban empat, tanpa gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.7 dapat disimpulkan bahwa pemakaian daya elektrik sebesar 3,468 Watt dengan waktu keadaan mantap (t_s) selama 23 detik, *error steady state* 1,4 %, waktu naik (t_r) selama 13 detik, waktu tunda (t_d) selama 6 detik, waktu puncak (t_p) selama 14 detik dan maksimum *overshoot* (M_p) 3,533 %.

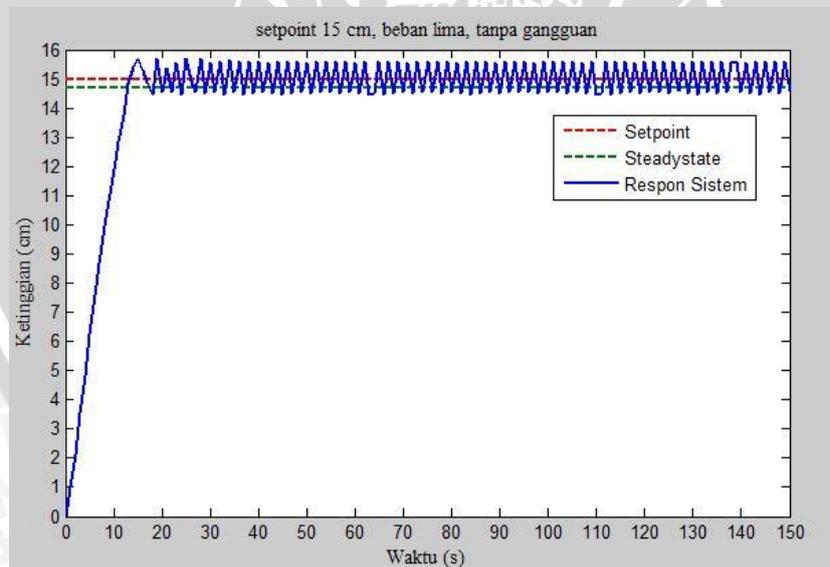
h) Pengujian pada *setpoint* 15 cm, beban empat, dengan gangguan.



Gambar 4.8 Grafik respon sistem pada *setpoint* 15 cm, beban empat, dengan gangguan.

Pada grafik respon sistem seperti yang ditunjukkan dalam Gambar 4.8 dengan adanya gangguan selama 38 detik dapat disimpulkan bahwa pemakaian energi elektrik sebesar 3,695 Watt dengan waktu keadaan mantap (t_s) selama 24 detik, *error steady state* 1,6 % dan maksimum *overshoot* (M_p) 3,66 %.

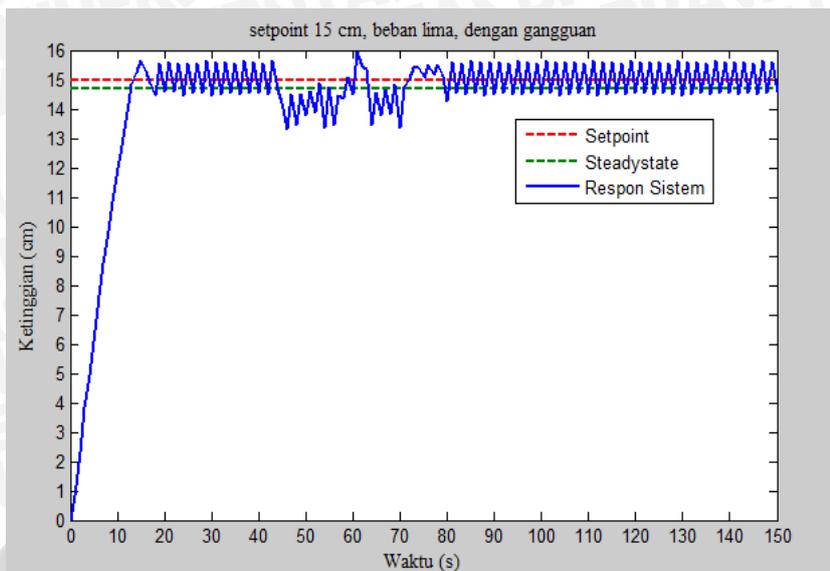
i) Pengujian pada *setpoint* 15 cm, beban lima, tanpa gangguan.



Gambar 4.9 Grafik respon sistem pada *setpoint* 15 cm, beban lima, tanpa gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.9 dapat disimpulkan bahwa pemakaian daya elektrik sebesar 4,124 Watt dengan waktu keadaan mantap (t_s) selama 26 detik, *error steady state* 1,8 %, waktu naik (t_r) selama 14 detik, waktu tunda (t_d) selama 7 detik, waktu puncak (t_p) selama 15 detik dan maksimum *overshoot* (M_p) 4,33 %.

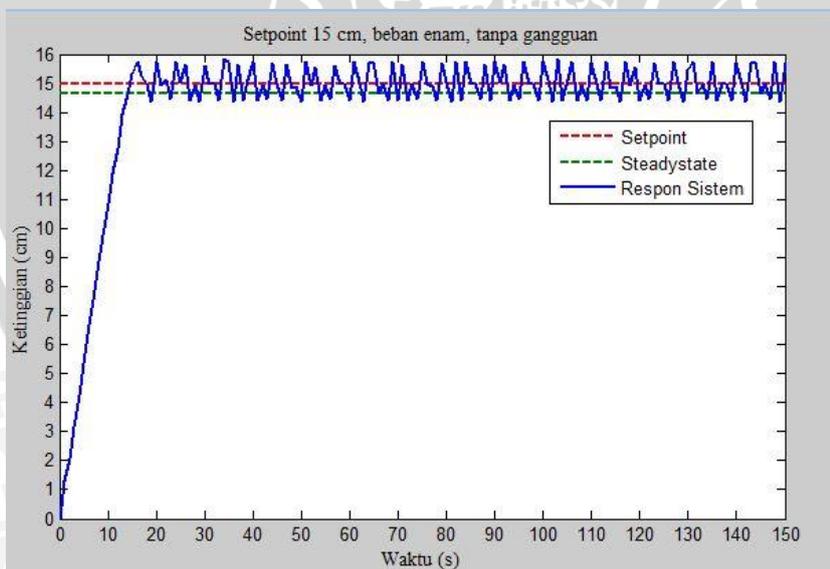
j) Pengujian pada setpoint 15 cm, beban lima, dengan gangguan.



Gambar 4.10 Grafik respon sistem pada *setpoint* 15 cm, beban lima, dengan gangguan.

Pada grafik respon sistem seperti yang ditunjukkan dalam Gambar 4.10 dengan adanya gangguan selama 36 detik dapat disimpulkan bahwa pemakaian energi elektrik sebesar 3,675 Watt dengan waktu keadaan mantap (t_s) selama 27 detik, *error steady state* 1.8 % dan maksimum *overshoot* (M_p) 4,46 %.

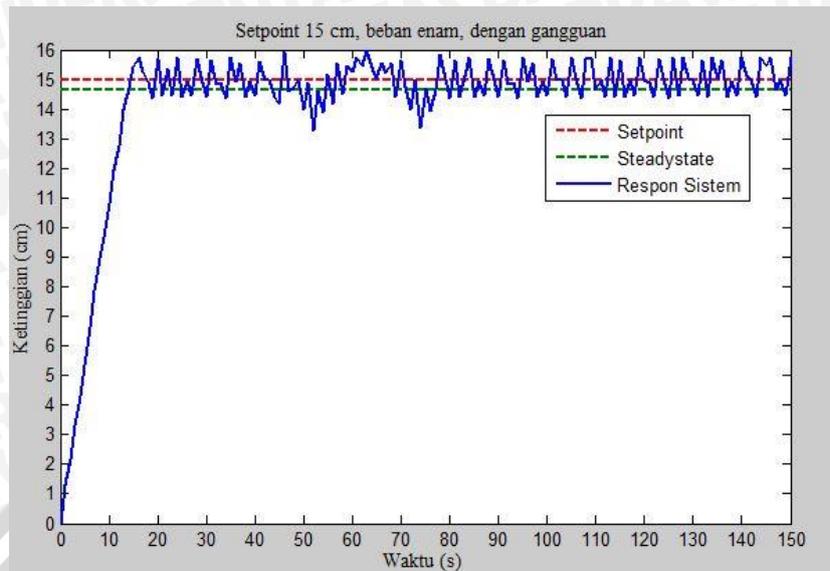
k) Pengujian pada *setpoint* 15 cm, beban enam, tanpa gangguan.



Gambar 4.11 Grafik respon sistem pada *setpoint* 15 cm, beban enam, tanpa gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.11 dapat disimpulkan bahwa pemakaian daya elektrik sebesar 4.891 Watt dengan waktu keadaan mantap (t_s) selama 28 detik, *error steady state* 2,06 %, waktu naik (t_r) selama 15 detik, waktu tunda (t_d) selama 8 detik, waktu puncak (t_p) selama 16 detik dan maksimum *overshoot* (M_p) 4,867 %.

1) Pengujian pada setpoint 15 cm, beban enam, dengan gangguan.



Gambar 4.12 Grafik respon sistem pada *setpoint* 15 cm, beban enam, dengan gangguan.

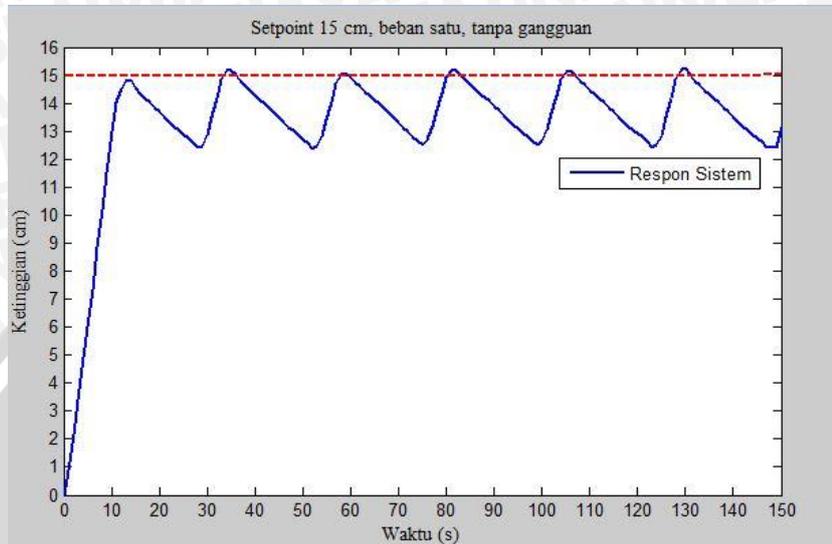
Pada grafik respon sistem seperti yang ditunjukkan dalam Gambar 4.12 dengan adanya gangguan selama 31 detik dapat disimpulkan bahwa pemakaian energi elektrik sebesar 4,373 Watt dengan waktu keadaan mantap (t_s) selama 30 detik, *error steady state* 2.2% dan maksimum *overshoot* (M_p) 5%.

Berdasarkan grafik respon sistem dengan variasi beban tanpa gangguan menunjukkan bahwa nilai pemakaian daya elektrik terus mengalami kenaikan. Sedangkan, semakin banyak variasi beban dengan gangguan maka nilai pemakaian daya elektrik mengalami penurunan seperti yang terlihat pada gambar grafik beban lima dan beban enam yang diberi gangguan.

4.2 Pengujian Sistem Menggunakan kontroler *On-Off*

Pengujian sistem dengan menggunakan kontroler *on-off* dilakukan untuk mengetahui respon sistem serta penggunaan pemakaian daya elektrik dengan variasi beban maupun variasi beban yang diberi gangguan.

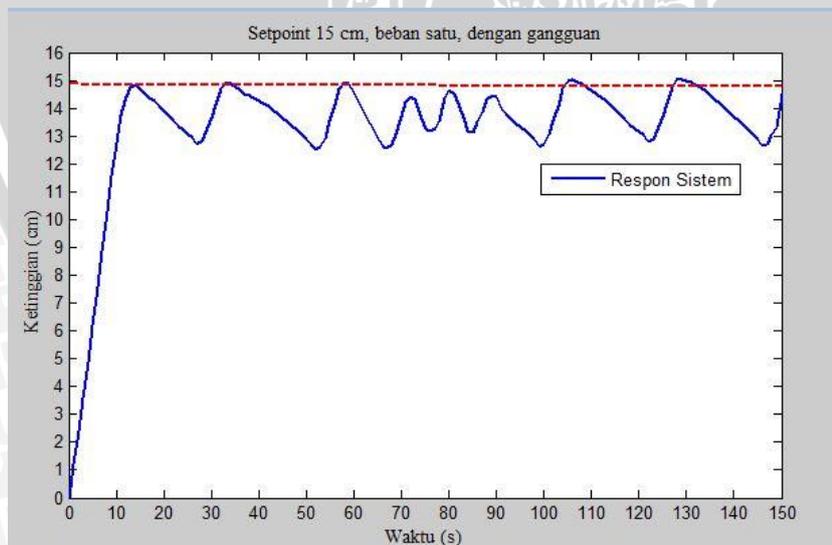
a) Pengujian pada *setpoint* 15 cm, beban satu, tanpa gangguan.



Gambar 4.13 Grafik respon sistem pada *setpoint* 15 cm, beban satu, tanpa gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4,13 dapat disimpulkan bahwa penggunaan pemakaian daya elektrik sebesar 1,335 Watt.

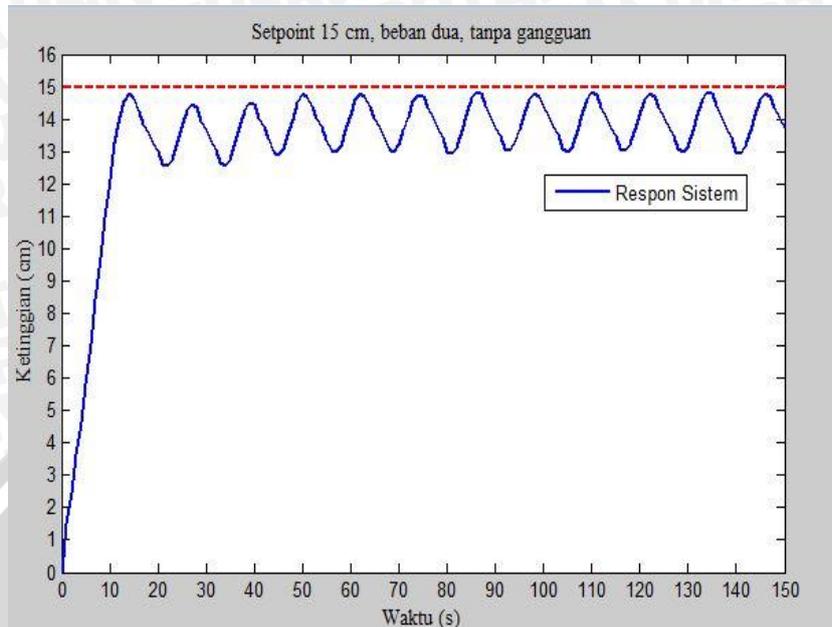
b) Pengujian pada *setpoint* 15 cm, beban satu, dengan gangguan.



Gambar 4.14 Grafik respon sistem pada *setpoint* 15 cm, beban satu, dengan gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.14 dapat disimpulkan bahwa pada saat diberi gangguan selama 41 detik maka diketahui penggunaan pemakaian daya elektrik sebesar 1,859 Watt.

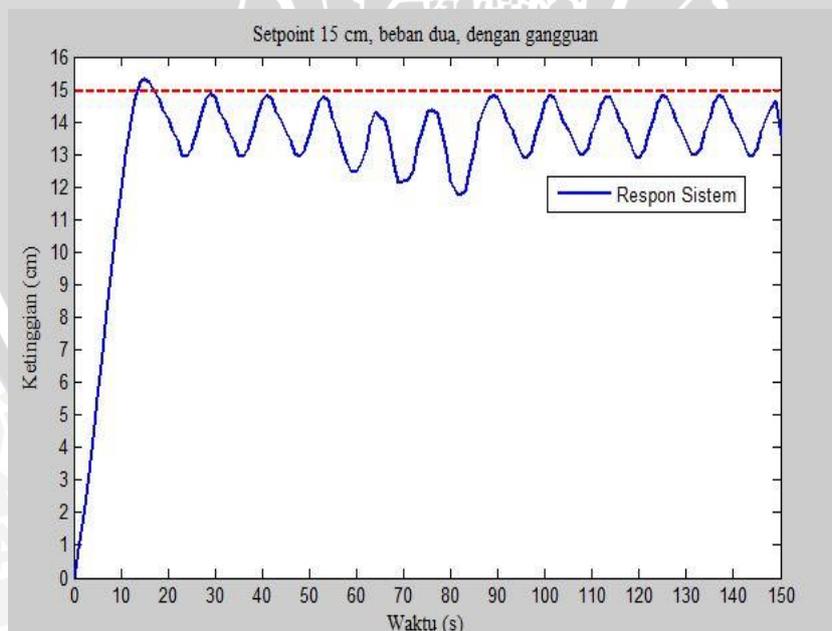
c) Pengujian pada *setpoint* 15 cm, beban dua, tanpa gangguan.



Gambar 4.15 Grafik respon sistem pada *setpoint* 15 cm, beban dua, tanpa gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.15 dapat disimpulkan bahwa penggunaan pemakaian daya elektrik sebesar 2,193 Watt.

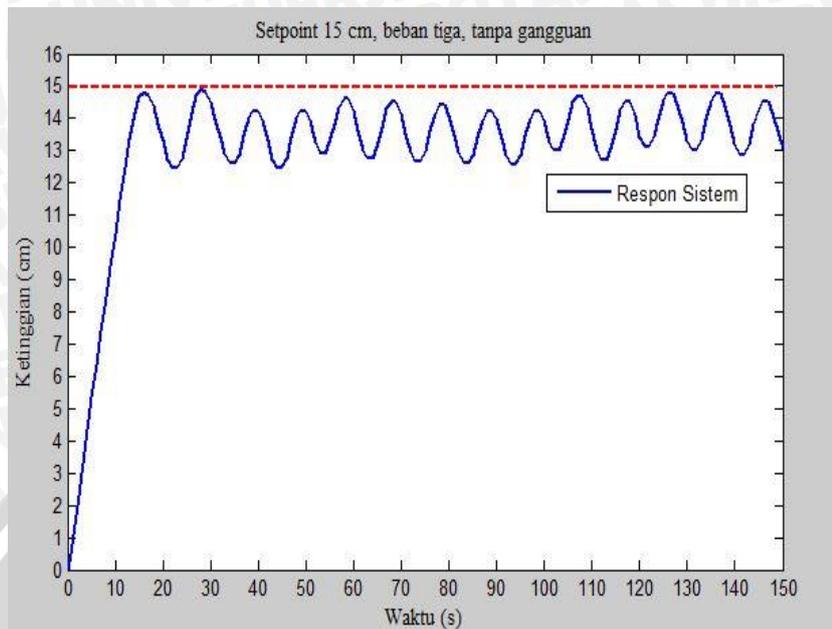
d) Pengujian pada *setpoint* 15 cm, beban dua, dengan gangguan.



Gambar 4.16 Grafik respon sistem pada *setpoint* 15 cm, beban dua, dengan gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.16 dapat disimpulkan bahwa pada saat diberi gangguan selama 23 detik maka diketahui penggunaan pemakaian daya elektrik sebesar 2,71 Watt.

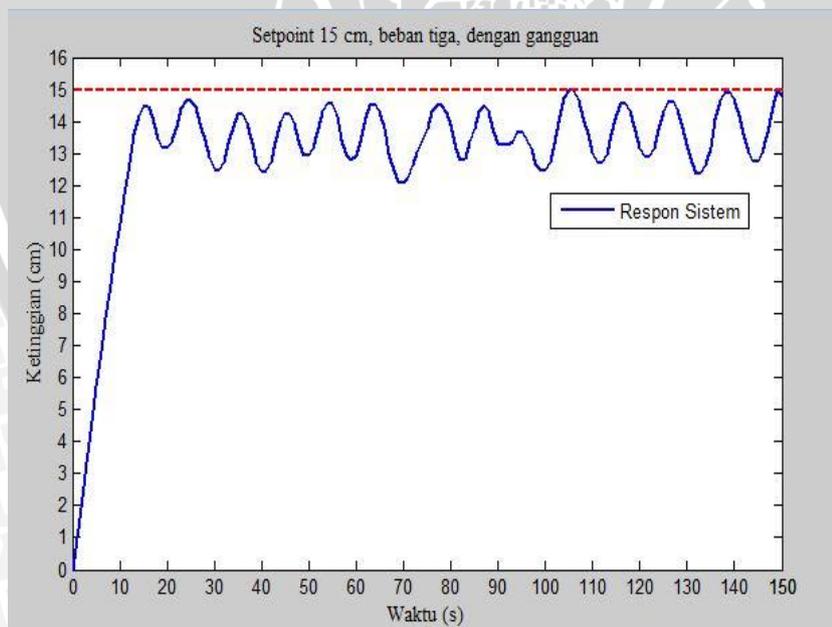
e) Pengujian pada *setpoint* 15 cm, beban tiga, tanpa gangguan.



Gambar 4.17 Grafik respon sistem dengan *setpoint* 15 cm, beban tiga, tanpa gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.17 dapat disimpulkan bahwa penggunaan pemakaian daya elektrik sebesar 3,29 Watt.

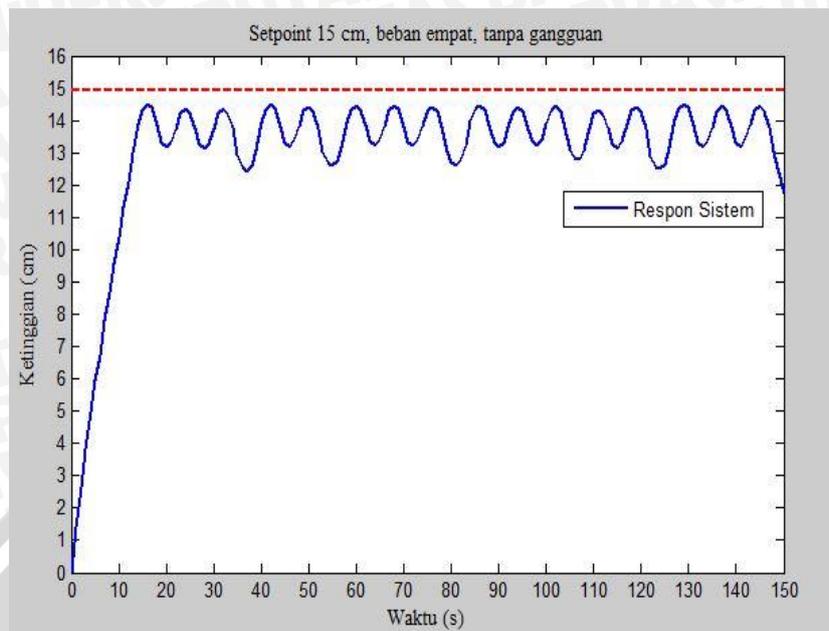
f) Pengujian pada *setpoint* 15 cm, beban tiga, dengan gangguan.



Gambar 4.18 Grafik respon sistem pada *setpoint* 15 cm, beban tiga, dengan gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.18 dapat disimpulkan bahwa pada saat diberi gangguan selama 35 detik maka diketahui penggunaan pemakaian daya elektrik sebesar 3,48 Watt.

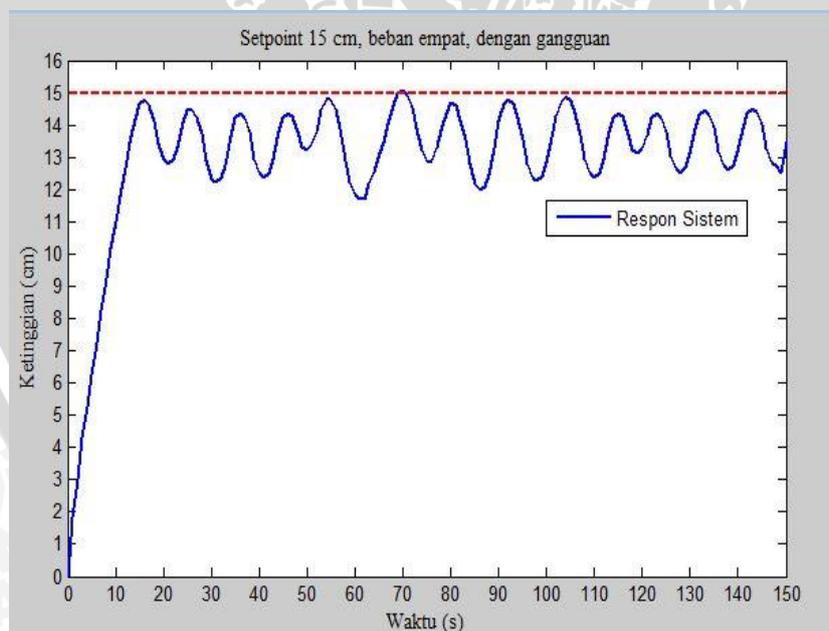
g) Pengujian pada *setpoint* 15 cm, beban empat, tanpa gangguan.



Gambar 4.19 Grafik respon sistem pada *setpoint* 15 cm, beban empat, tanpa gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.19 dapat disimpulkan bahwa penggunaan pemakaian daya elektrik sebesar 3,816 Watt.

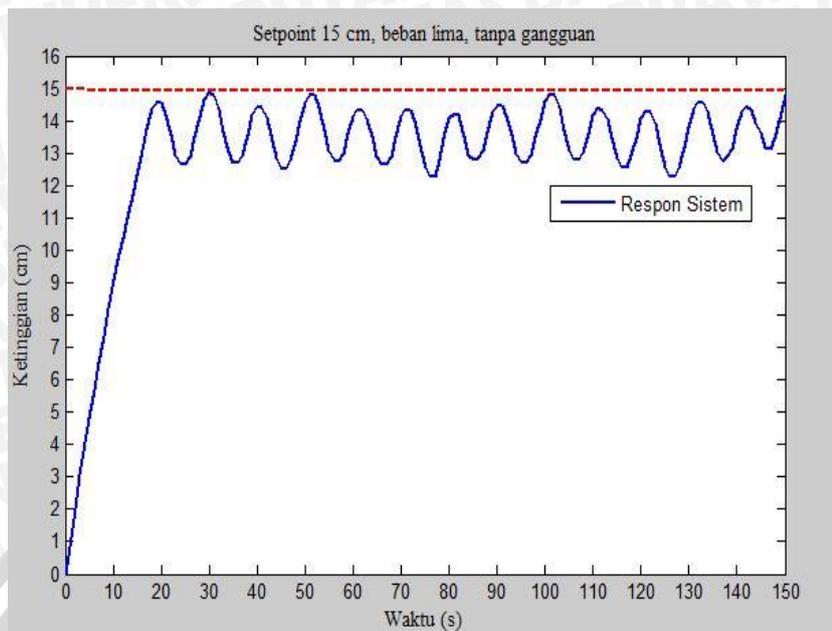
h) Pengujian pada *setpoint* 15 cm, beban empat, dengan gangguan.



Gambar 4.20 Grafik *output Plant* setpoint 15 cm, beban empat, dengan gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.20 dapat disimpulkan bahwa pada saat diberi gangguan selama 37 detik maka diketahui penggunaan pemakaian daya elektrik sebesar 3,909 Watt.

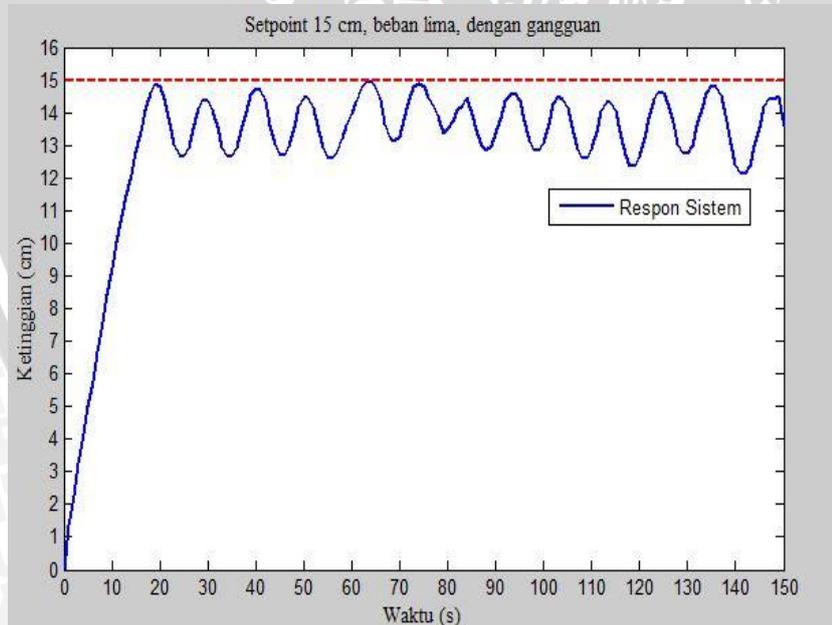
i) Pengujian pada *setpoint* 15 cm, beban lima, tanpa gangguan.



Gambar 4.21 Grafik *output Plant* setpoint 15 cm, beban lima, tanpa gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.21 dapat disimpulkan bahwa penggunaan pemakaian daya elektrik sebesar 4,72 Watt.

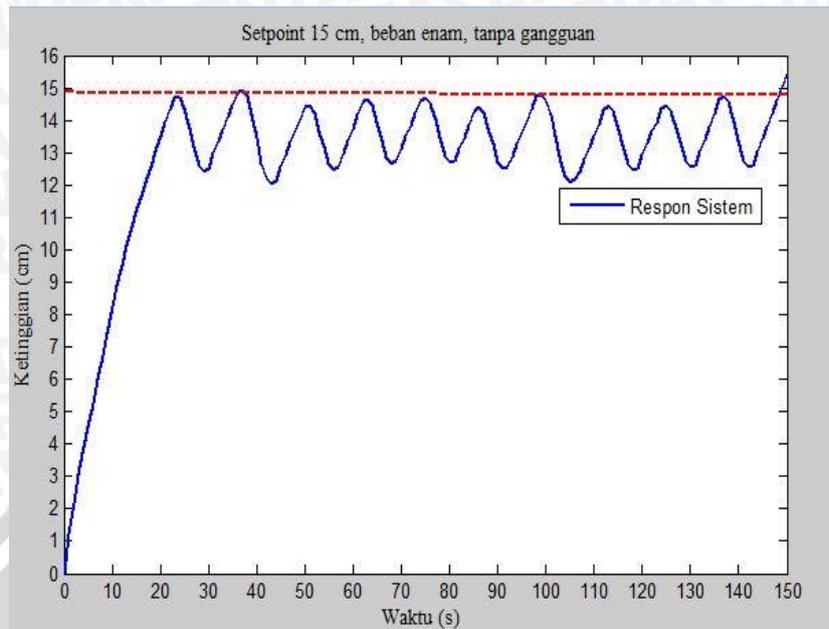
j) Pengujian pada *setpoint* 15 cm, beban lima, dengan gangguan.



Gambar 4.22 Grafik respon sistem pada *setpoint* 15 cm, beban lima, dengan gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.22 dapat disimpulkan bahwa pada saat diberi gangguan selama 32 detik maka diketahui penggunaan pemakaian daya elektrik sebesar 4,529 Watt.

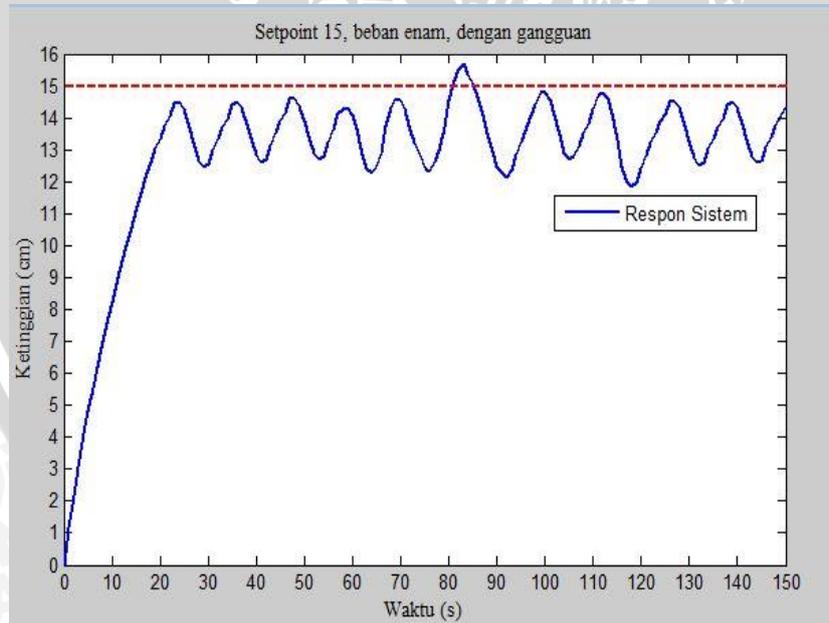
k) Pengujian pada *setpoint* 15 cm, beban enam, tanpa gangguan.



Gambar 4.23 Grafik respon sistem pada *setpoint* 15 cm, beban enam, tanpa gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.23 dapat disimpulkan bahwa penggunaan pemakaian daya elektrik sebesar 5,626 Watt.

l) Pengujian pada *setpoint* 15 cm, beban enam, dengan gangguan.



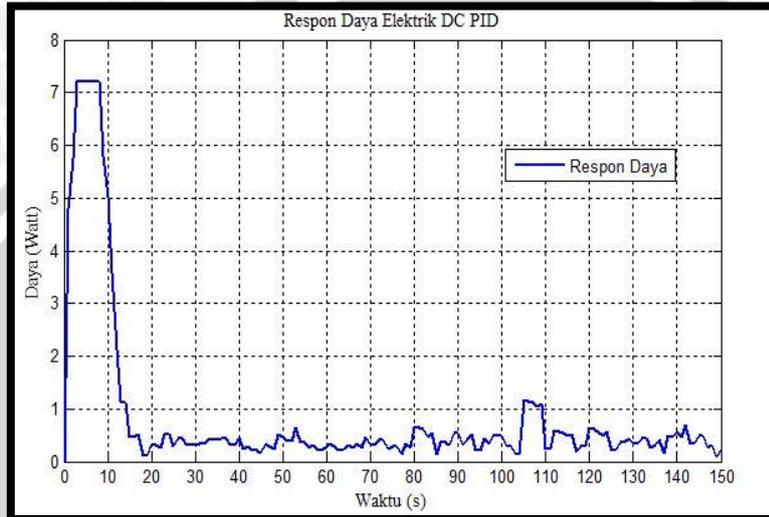
Gambar 4.24 Grafik respon sistem *setpoint* 15 cm, beban enam, dengan gangguan.

Pada grafik respon sistem yang ditunjukkan dalam Gambar 4.24 dapat disimpulkan bahwa pada saat diberi gangguan selama 39 detik maka diketahui penggunaan pemakaian daya elektrik sebesar 5,006 Watt.

Berdasarkan grafik respon sistem dengan variasi beban tanpa gangguan menunjukkan bahwa nilai pemakaian daya elektrik terus mengalami kenaikan. Sedangkan, semakin banyak variasi beban dengan gangguan maka nilai pemakaian daya elektrik mengalami penurunan.

4.3 Respon Daya Elektrik dengan Kontroler PID

Berikut merupakan salah-satu hasil pengujian respon daya elektrik dengan kontroler PID.

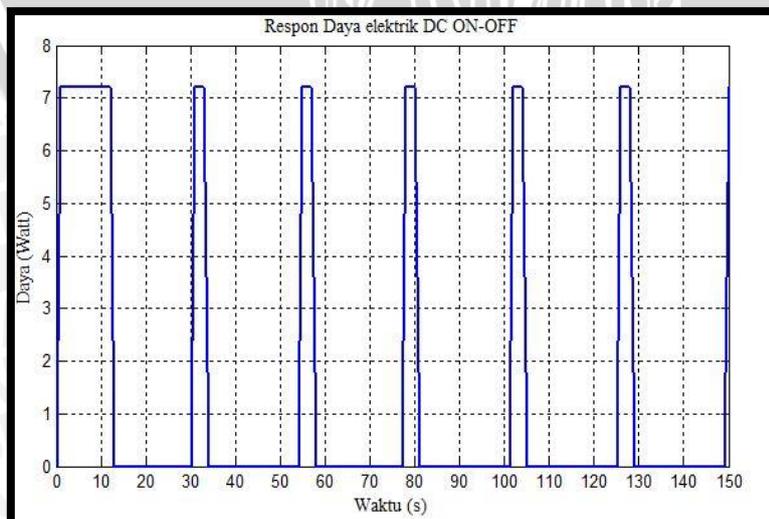


Gambar 4.25 Grafik respon daya elektrik dengan kontroler PID

Pada grafik yang ditunjukkan dalam Gambar 4.25 dapat disimpulkan, bahwa respon daya dengan kontroler PID dapat menyesuaikan tingkat level ketinggian air di dalam tandon.

4.4 Respon Daya Elektrik dengan Kontroler *On-Off*

Berikut merupakan salah-satu hasil pengujian respon daya elektrik dengan kontroler *on-off*



Gambar 4.26 Grafik respon daya elektrik dengan kontroler *on-off*

Pada Gambar 4.26 grafik respon dengan kontroler *on-off* dapat disimpulkan, setiap kurang dari batas atas dan lebih dari batas bawah ($15 \text{ cm} \leq \text{ketinggian} \geq 12.5 \text{ cm}$) pengisian air membutuhkan daya maksimal agar level ketinggian air di dalam tandon cepat mencapai batas atas berupa *setpoint* sebesar 15 cm, setelah mencapai batas atas maka level ketinggian air akan turun mencapai batas bawah sebesar 12,5 cm dengan daya minimal yaitu 0 watt dan akan melakukan pengisian air kembali dengan daya maksimal seperti langkah sebelumnya.





BAB V PENUTUP

5.1 Kesimpulan

Pada pengontrolan level ketinggian air pada tandon untuk gedung bertingkat guna pemakaian daya elektrik dengan menggunakan metode *Ziegler-nichols* 1 didapatkan nilai parameter PID sebesar $K_p = 20.19$, $K_i = 43.4$, $K_d = 2.3$ sehingga dapat mempercepat sistem untuk mencapai keadaan *steadystate* sesuai dengan *setpoint*. Pada variasi beban 1-6 tanpa gangguan didapatkan waktu keadaan mantap (t_s) selama 14 detik, 18 detik, 20 detik, 23 detik, 26 detik, 28 detik, *error steady state* 0.467 %, 0.9%, 1.167%, 1.4%, 1.8%, 2.06%, dan maksimum *overshoot* (M_p) 1.7333 %, 2.467%, 3.133%, 3.533%, 4.33%, 4.867%, serta pada variasi beban 1-6 dengan gangguan didapatkan waktu keadaan mantap (t_s) selama 15 detik, 19 detik, 21 detik, 24 detik, 27 detik, 30 detik, *error steady state* 0.534%, 1%, 1.3%, 1.6%, 1.8%, 2.2%, dan maksimum *overshoot* (M_p) 1.866%, 2.6%, 3.266%, 3.66%, 4.46%, 5%. Sedangkan pada kontroler *on-off* hanya memiliki dua keadaan yakni batas atas dan batas bawah yang membuat sistem tidak dapat mencapai keadaan *steadystate* sesuai dengan *setpoint*.

Pada perhitungan pemakaian daya elektrik diketahui bahwa penggunaan kontroler PID lebih hemat dibandingkan dengan kontroler *on-off* baik menggunakan variasi beban maupun variasi beban dengan gangguan.

5.2 Saran

Dalam penelitian yang saya lakukan ini masi berupa Tegangan dan Arus DC sehingga menghasilkan pemakaian Daya Elektrik DC, maka saran yang dapat saya berikan untuk penelitian selanjutnya adalah dengan mengembangkan penelitian saya menggunakan penerapan Tegangan dan Arus AC sehingga didapatkan penerapan Daya secara nyata.



DAFTAR PUSTAKA

- Sugeng, Yusril. 2015. *Sistem Kendali Air Bersih Pada Gedung Bertingkat*, Jurnal, Universitas Islam 45 Bekasi.
- Thiang, Yohanes TDS, Andre Mulya. 2005. *Pengaturan Level Ketinggian Air Menggunakan Kontrol PID*, Jurnal, Fakultas Teknologi Industri, Jurusan Teknik Elektro, Universitas Kristen Petra Surabaya.
- Ogata, K. 1997. *Teknik Kontrol Automatik Jilid 2*. Jakarta: Penerbit Erlangga.
- Prasetyo Joko. 2016. *Uji Performansi Pada Sistem Kontrol Level Air Dengan Variasi Beban Menggunakan Kontroler PID*. Malang: Skripsi Teknik Elektro Universitas Brawijaya Malang.
- Simamora Kristianingsih. 2015. *Desain Kendali PID Pada Plant Debit Air Dengan Metode Ziegler-nichols dan Cohen-Coon Menggunakan Matlab dan Arduino*. Bandung: Laporan Teknik Elektronika Jurusan Teknik Elektro Politeknik Negeri.
- Jufri Hilman. 2012. *Rancang Bangun Alat Ukur Daya Arus Bolak-Balik Berbasis Mikrokontroler Atmega 8535*, Jurnal, Fakultas Mipa, Jurusan Fisika, Universitas Sumatera Utara.
- Haryanto Dwi. 2013. *Daya dalam fisika yang merupakan laju energi yang dihantarkan atau kerja yang dilakukan per satuan waktu. Daya dilambangkan dengan P*.
<http://riandyerlangga.ilearning.me/2013/09/11/hubungan-daya-arus-tegangan/>
- Rahman Arief. 2014. *Pengendalian ketinggian air pada distilasi air laut menggunakan kontroler On-Off*, Jurnal, Fakultas, Jurusan Teknik Elektro Universitas Brawijaya Malang.
- Iskandar Dede. 2012. *Kendali Level Air Dengan Menggunakan Parameter PID Kontroler*, Jurnal, Fakultas Teknik, Jurusan Teknik Elektro, Universitas Jendral Ahmad Yani Cimahi.
- Simamora Kristianingsih. 2015. *Desain Kendali PID Pada Plant Debit Air Dengan Metode Ziegler-nichols Menggunakan Matlab dan Arduino*, Jurnal, Fakultas Teknik, Jurusan Teknik Elektronika, Politeknik Negeri Bandung.

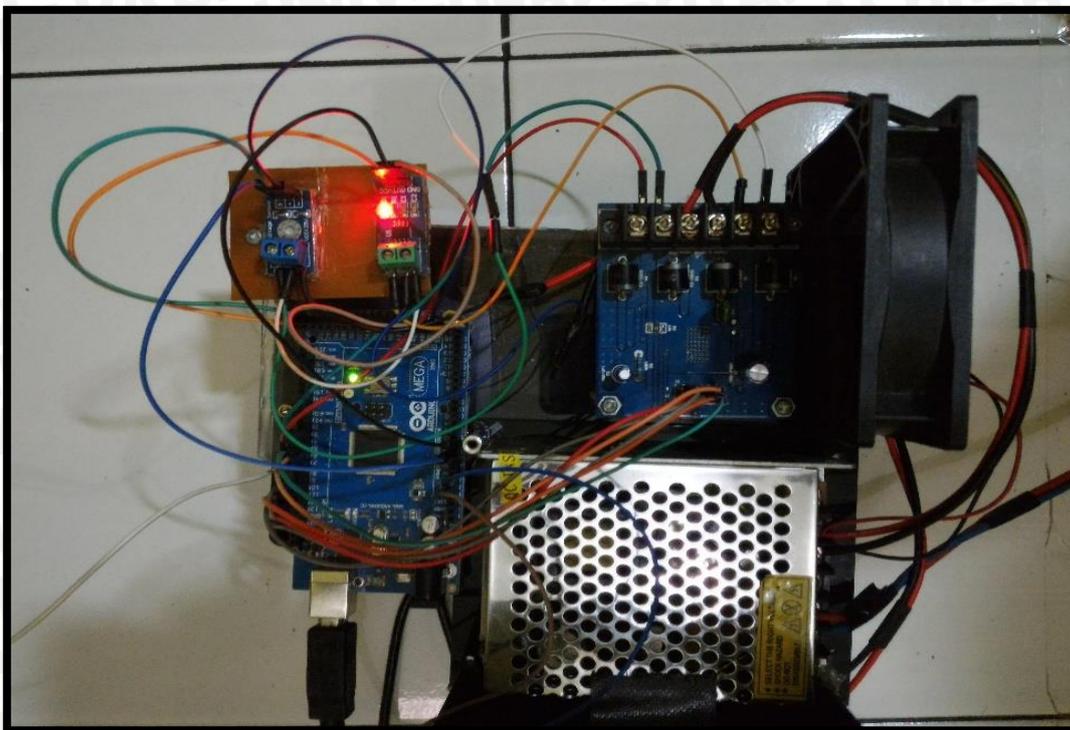


UNIVERSITAS BRAWIJAYA



LAMPIRAN I

Foto Alat



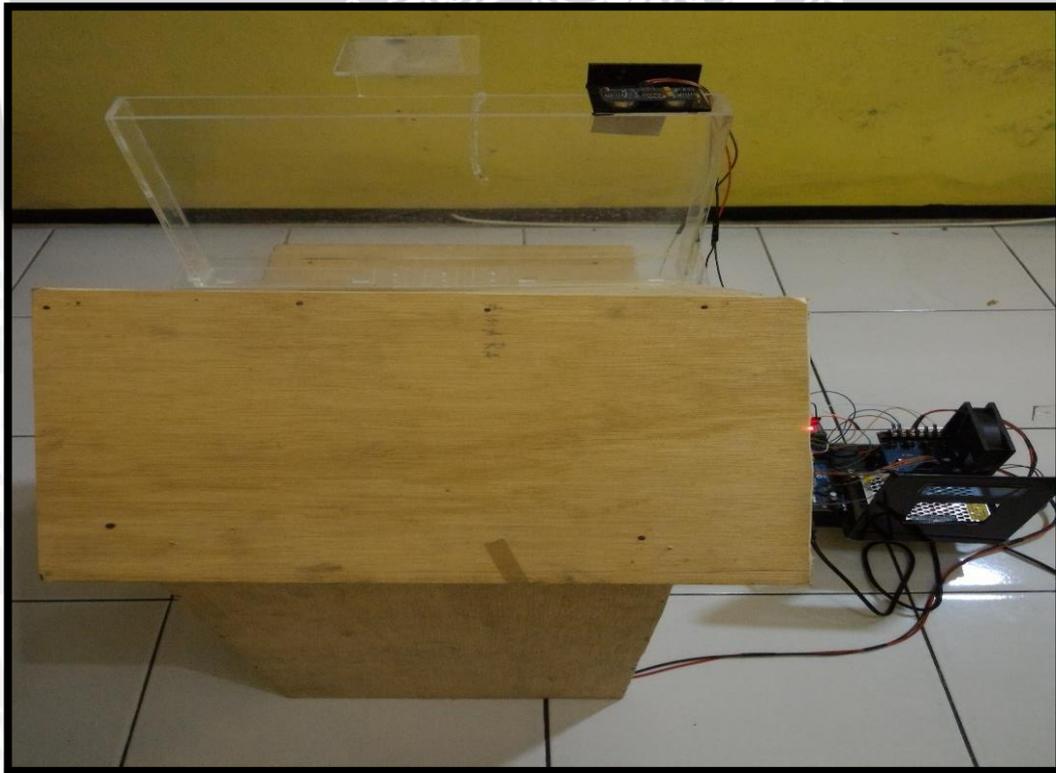
Gambar Keseluruhan Alat



Gambar *Plant* tampak atas

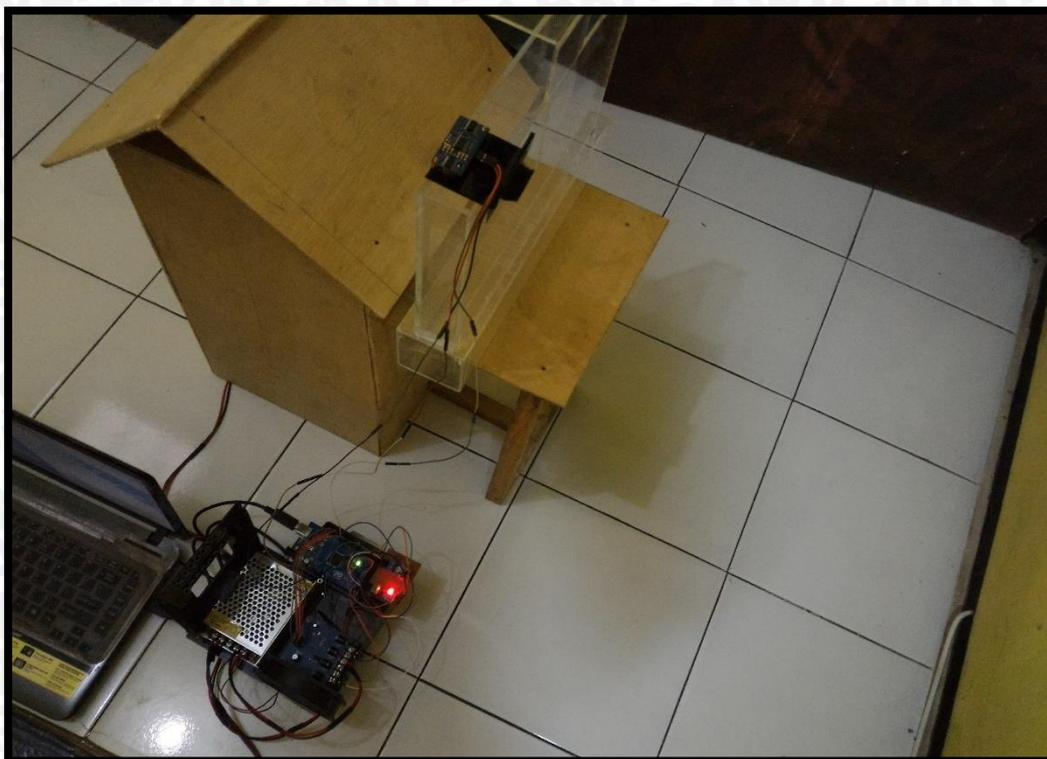


Gambar *Plant* tampak samping

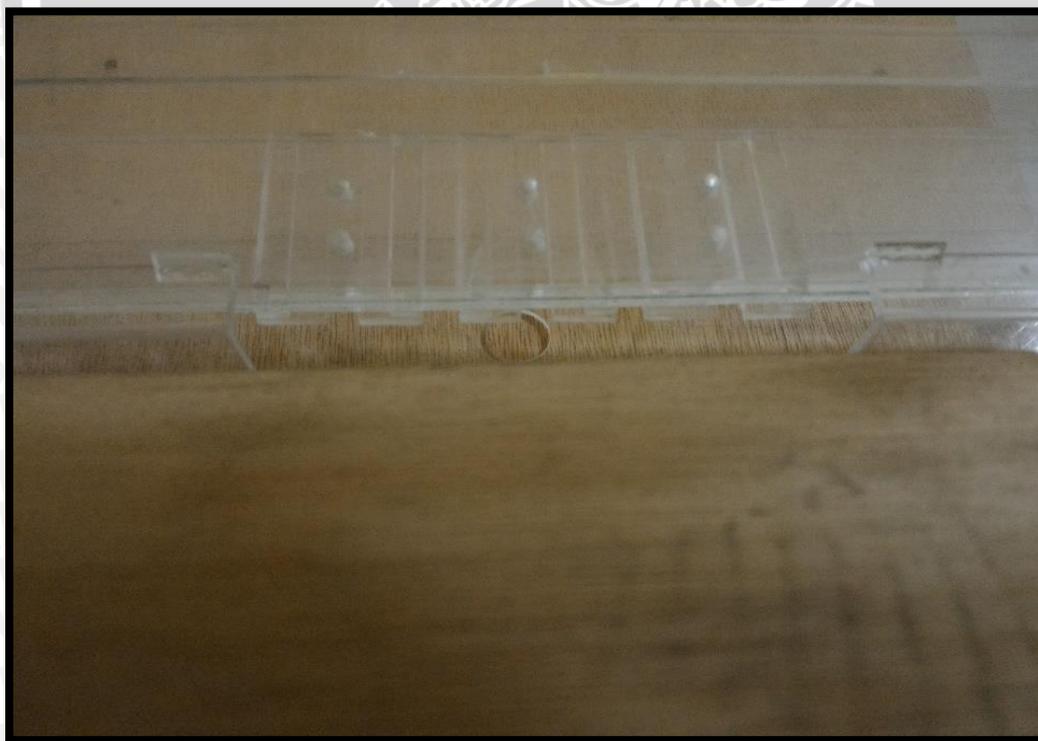


Gambar *Plant* tampak depan

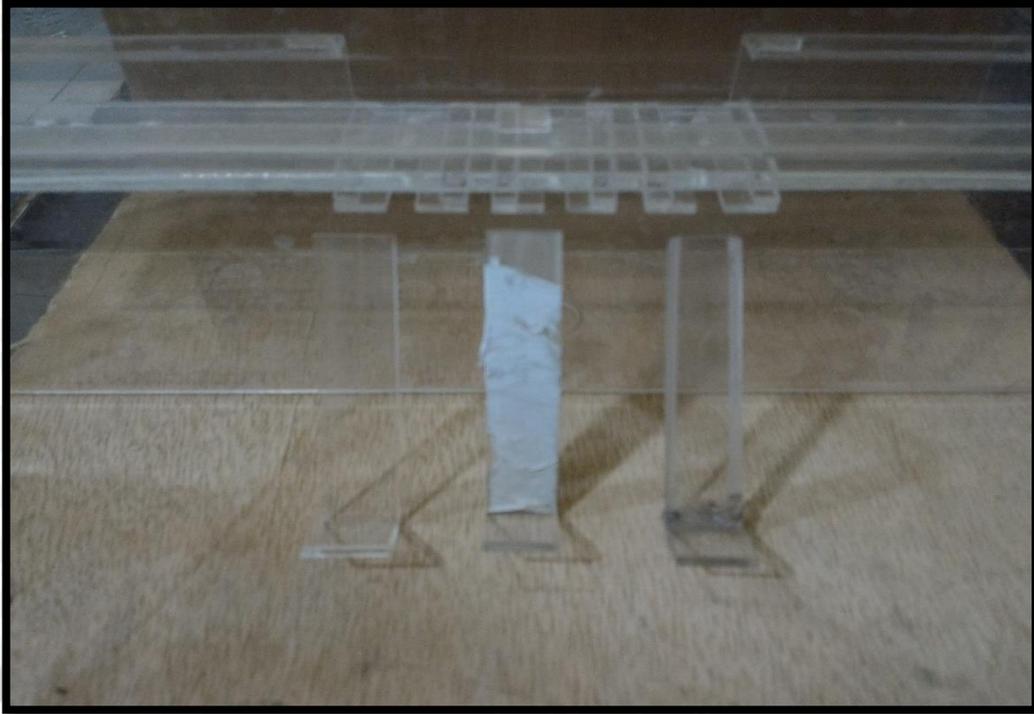




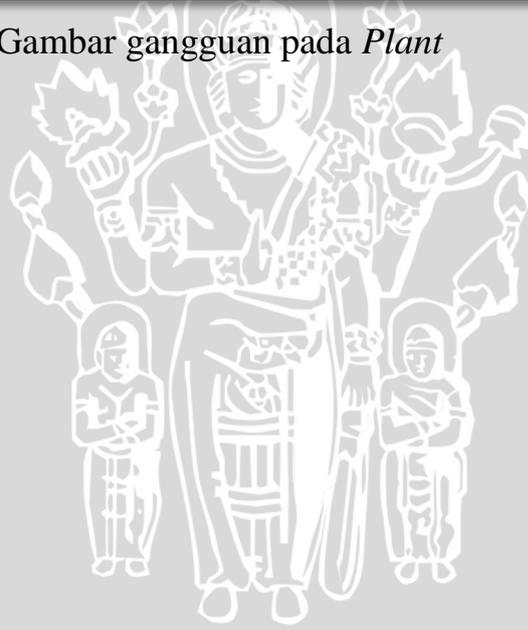
Gambar *Plant* tampak belakang



Gambar lubang keluaran air



Gambar gangguan pada *Plant*





UNIVERSITAS BRAWIJAYA



LAMPIRAN II

Listing Program

Arduino Program

```
/******
```

```
/* KEMAS MUHAMMAD RIZAL. */
```

```
*****
```

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

```
double Setpoint, Input, Output;
```

```
// TUNING PARAMETER
```

```
float Kp=20;
```

```
float Ki=43;
```

```
float Kd=2.32;
```

```
// INISIALISASI
```

```
float nilaisensorarus;
```

```
float nilaisensortegangan;
```

```
float arus;
```

```
float tegangan;
```

```
float daya;
```

```
float terukur;
```

```
float pingpin = 7;
```

```
float duration;
```

```
float cm;
```

```
float pwm;
```

```
float dutycycle;
```

```
float k=0.39;
```



```
// TENTUKAN LINK DAN PARAMETER PENYETEL AWAL
```

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

```
// TIMESTAMP
```

```
unsigned long serialTime; //this will help us know when to talk with processing
```

```
const int sampleRate = 1; // Variable that determines how fast our PID loop runs
```

```
const long serialPing = 500; //This determines how often we ping our loop
```

```
unsigned long now = 0; //This variable is used to keep track of time
```

```
unsigned long lastMessage = 0; //This keeps track of when our loop last spoke to serial
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  //PIN DRIVER MOTOR EMS 5A
```

```
  pinMode(13,OUTPUT); //VCC driver, driver pin 7
```

```
  pinMode(12,OUTPUT); //GRD driver, driver pin 8
```

```
  pinMode(6,OUTPUT); //MIN1 VCC motor1 PWM, driver pin 1
```

```
  pinMode(5,OUTPUT); //MIN2 GRD motor1 PWM, driver pin 2
```

```
  pinMode(10,OUTPUT); //MEN, driver pin 4
```

```
  pinMode(9,OUTPUT); //MSLP, driver pin 6
```

```
  pinMode(4,OUTPUT); //VCC CATU DRIVER
```

```
  pinMode(3,OUTPUT); //GRD DRIVER
```

```
  digitalWrite(13, HIGH);
```

```
  digitalWrite(12, LOW);
```

```
  digitalWrite(6, HIGH);
```

```
  digitalWrite(5, LOW);
```

```
  digitalWrite(10, HIGH);
```

```
  digitalWrite(9, HIGH);
```

```
digitalWrite(4, HIGH);
```

```
digitalWrite(3, LOW);
```

```
//PIN SENSOR ARUS
```

```
pinMode(11,OUTPUT); //VCC SENSOR ARUS
```

```
digitalWrite(11, HIGH);
```

```
//PIN SENSOR TEGANGAN
```

```
pinMode(8,OUTPUT); //VCC SENSOR TEGANGAN
```

```
pinMode(2,OUTPUT); //GRD SENSOR TEGANGAN
```

```
digitalWrite(8, HIGH);
```

```
digitalWrite(2, LOW);
```

```
//INISIALISASI PARAMETER
```

```
Input = terukur;
```

```
Setpoint = 15;
```

```
//INISIALISASI PID
```

```
myPID.SetMode(AUTOMATIC); // Turn on the PID loop as automatic control
```

```
myPID.SetOutputLimits(0, 999);
```

```
myPID.SetSampleTime(sampleRate); // Sets the sample rate
```

```
lastMessage = millis(); // timestamp
```

```
}
```

```
void loop()
```

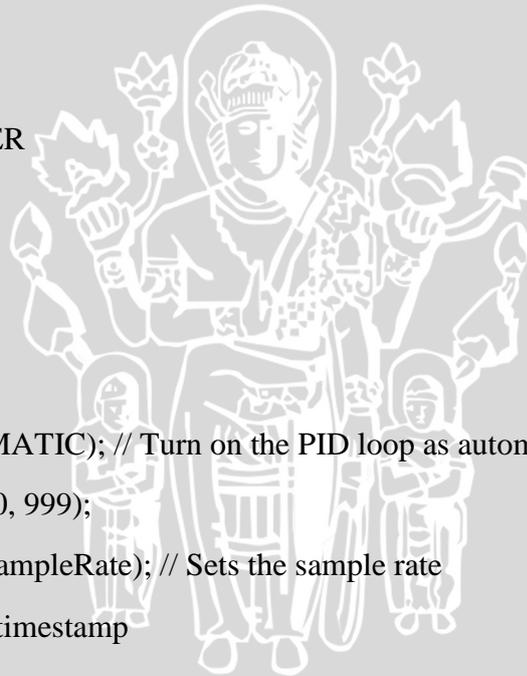
```
{
```

```
//INISIALISASI SENSOR PING
```

```
pinMode(pingpin, OUTPUT);
```

```
digitalWrite(pingpin, LOW);
```

```
delayMicroseconds(2);
```



```
digitalWrite(pingpin, HIGH);  
delayMicroseconds(5);  
digitalWrite(pingpin, LOW);  
pinMode(pingpin, INPUT);  
duration = pulseIn(pingpin, HIGH);  
cm = duration / 58;  
terukur = 20 - cm;
```

```
//INISIALISASI SENSOR ARUS
```

```
nilaisensorarus=analogRead(A0);  
arus=((nilaisensorarus-488)*27.03 / 1023)-1;
```

```
//INISIALISASI SENSOR TEGANGAN
```

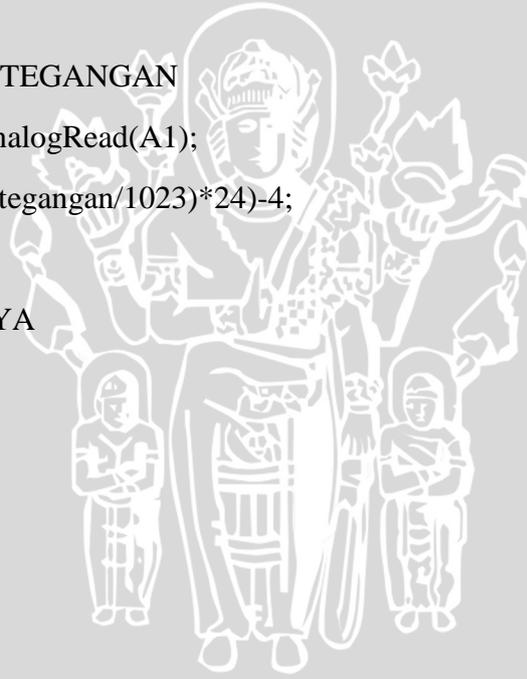
```
nilaisensortegangan=analogRead(A1);  
tegangan=((nilaisensortegangan/1023)*24)-4;
```

```
//RUMUS MENCARI DAYA
```

```
daya=tegangan*arus;
```

```
//SERIAL MONITOR
```

```
Serial.print(arus);  
Serial.print("\t\t");  
Serial.print(tegangan);  
Serial.print("\t\t");  
Serial.print(daya);  
Serial.print("\t\t");  
Serial.println(terukur);
```



```
//KALIBRASI
```

```
Input = terukur;
```

```
myPID.Compute();
```

```
dutycycle = Output/k;
```

```
pwm=(dutycycle * 2.55);
```

```
if (pwm>255){pwm=255;}
```

```
analogWrite(6, pwm);
```

```
now = millis();
```

```
delay(2000);
```

```
}
```



UNIVERSITAS BRAWIJAYA



LAMPIRAN III

Datasheet