

BAB I PENDAHULUAN

1.1 Latar Belakang

Kontroler merupakan salah satu komponen sistem kontrol yang berfungsi mengolah sinyal umpan balik dan sinyal masukan acuan (*setpoint*) atau sinyal *error* menjadi sinyal kontrol (Ogata K.,1997) agar dapat menghasilkan *output* sistem yang sesuai dengan sinyal masukan acuan dan memiliki *error steady state*, *overshoot* dan *time settling* yang rendah. Ada beberapa jenis kontroler diantaranya kontroler dua posisi (*on-off*), kontroler proporsional, kontroler integral, kontroler diferensial, dan kontroler proporsional integral diferensial (PID). Kontroler PID sudah ada sejak dua pertiga abad yang lalu dan menjadi komponen kunci dalam kontrol proses industri (Rozsa L.,1990). Kontroler PID sendiri terdapat 2 jenis yaitu analog dan digital. Kontroler PID analog menggunakan perangkat seperti penguat operasional, kapasitor dan resistor yang dapat diandalkan dalam sistem umpan balik mekanis. Pada era yang modern ini sistem pengontrolan sudah mencapai sistem yang kompleks, sehingga kontroler PID analog kurang cocok untuk sistem yang kompleks. Selain itu, daya tahannya terhadap gangguan eksternal kurang (Bibbero RJ.,1977).

Kontroler PID digital yang menggunakan mikrokontroler mendominasi kontrol industri saat ini. Pada mikrokontroler sudah terdapat *central processing unit* (CPU) sebagai tempat pemrosesan fungsi aritmatika, memori sebagai tempat penyimpanan data atau fungsi-fungsi, dan modul input output. Keuntungan dari kontroler berbasis mikrokontroler diantaranya harganya yang lebih murah dibanding dengan kontroler analog, mampu menggunakan algoritma kontrol yang kompleks, bobot lebih ringan, dan lebih kebal terhadap gangguan eksternal (Fassois S.,1982).

Plant tangki air 73426 di Laboratorium Sistem Kontrol Universitas Brawijaya saat ini menggunakan kontroler PID analog, kontroler tersebut mampu mengendalikan *plant* dengan baik. Hampir seluruh industri di dunia memanfaatkan teknologi kontrol yang serba digital karena kontrol digital memiliki beberapa kelebihan dibanding dengan

kontrol analog, diantaranya dapat digunakan untuk algoritma sistem yang kompleks, bobot yang lebih ringan karena meminimalkan penggunaan perangkat elektrik, lebih kebal terhadap gangguan eksternal atau *noise* yang biasa timbul di rangkaian elektrik konvensional, menampilkan data hasil dari pengontrolan serta mudah dalam mengubah parameter yang dibutuhkan kontroler. Berdasarkan uraian di atas, maka perlu dilakukan pengembangan pada *plant* tangki air 73426 menggunakan perangkat kontrol digital untuk mengimbangi perkembangan teknologi, sehingga dalam skripsi ini dirancang sebuah perangkat kontrol digital yang mampu mengendalikan ketinggian air pada *plant* tangki air 73426 di Laboratorium Sistem Kontrol Universitas Brawijaya.

1.2 Rumusan Masalah

1. Bagaimana merancang kontroler PID digital sebagai pengendali ketinggian air pada *plant* tangki air?
2. Bagaimana mendapatkan parameter kontrol PID sebagai pengendali ketinggian air dari hasil perancangan?
3. Bagaimana hasil perancangan kontroler PID digital sebagai pengendali ketinggian air pada *plant* tangki air?

1.3 Batasan Masalah

1. Pembahasan ditekankan pada penggunaan kontrol PID pada sistem.
2. Pengaturan yang dilakukan adalah pengaturan ketinggian air melalui pengaturan kecepatan putaran motor pompa.
3. Motor pompa yang digunakan adalah motor pompa DC dengan catu daya 12 V.
4. Kinerja *driver* dan elektronika tidak dibahas secara mendalam.
5. Sensor yang digunakan adalah sensor ultrasonic HC-SR04.
6. *Plant* yang digunakan adalah *plant* tangki air 73426 di Laboratorium Sistem Kontrol Teknik Elektro Universitas Brawijaya.

1.4 Tujuan

Penelitian ini bertujuan untuk merancang sistem pengendali ketinggian air pada *plant* tangki air dan mendapatkan parameter kontrol PID dari hasil perancangan

1.5 Sistematika Penulisan

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

BAB I Pendahuluan

Membahas latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan.

BAB II Tinjauan Pustaka

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

BAB III Metode Penelitian

Membahas metode penelitian dan perencanaan alat.

BAB IV Perancangan dan Pembuatan Alat

Membahas perancangan alat yang meliputi spesifikasi, perencanaan blok diagram, prinsip kerja dan pembuatan alat, serta bagaimana penerapannya dalam sistem secara keseluruhan.

BAB V Pengujian dan Analisis

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

BAB VI Penutup

Membahas kesimpulan perancangan dan saran-saran yang diperlukan untuk pengembangan selanjutnya.



BAB II

TINJAUAN PUSTAKA

2.1 *Plant* Tangki air (73426)

Plant tangki air (73426) terdiri atas beberapa bagian diantaranya *filling tank*, tangki penampungan, pompa, dan rangkaian sensor. *Filling tank* berbentuk balok yang pada bagian atasnya terdapat lubang sehingga membentuk sebuah ruang di dalamnya. Di dalam tangki itu sendiri disediakan tempat untuk menyelipkan sekat berlubang. Sekat ini akan memungkinkan alternatif bentuk perilaku *plant*. Pada bagian atas dan bawah tangki terdapat sebuah lubang untuk tempat masuk dan keluarnya air dari dan ke tangki penampungan. Untuk mengalirkan air dari tangki penampungan digunakan sebuah pompa. Gambar 2.1 menunjukkan bentuk dari tangki pengisian (*filling tank*) (Modul Praktikum Sistem Kontrol).



Gambar 2.1 Tangki Pengisian (*Filling Tank*)

2.1.1 Pompa Motor *Direct Current* (DC)

Pompa adalah suatu alat atau mesin yang digunakan untuk memindahkan cairan dari suatu tempat ke tempat yang lain melalui suatu media perpipaan dengan cara menambahkan energi pada cairan yang dipindahkan dan berlangsung secara terus menerus. Pompa beroperasi dengan prinsip membuat perbedaan tekanan antara bagian

masuk (*suction*) dengan bagian keluar (*discharge*). Dengan kata lain, pompa berfungsi mengubah tenaga mekanis dari suatu sumber tenaga (penggerak) menjadi tenaga kinetis (kecepatan). Tenaga ini berguna untuk mengalirkan cairan dan mengatasi hambatan yang ada sepanjang pengaliran. Pada skripsi ini digunakan pompa motor DC dengan catu daya sebesar 12 V yang ditunjukkan dalam Gambar 2.2



Gambar 2.2 Pompa Motor DC 12 V

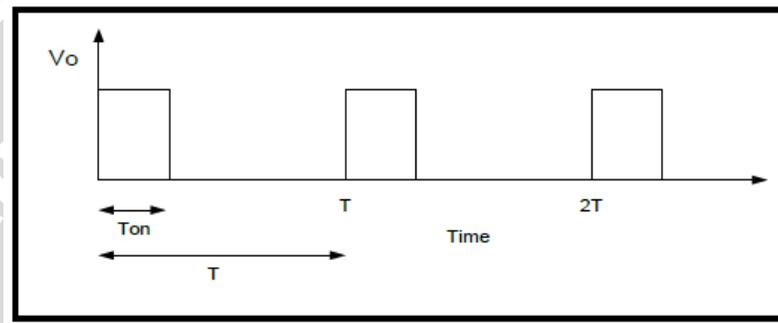
Prinsip kerja motor DC sesuai dengan hukum kemagnetan Lorentz, yaitu membangkitkan fungsi magnet pada suatu konduktor berarus dalam medan magnet sehingga timbul GGL induksi. Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kiri. Kaidah tangan kiri untuk motor menunjukkan arah arus yang mengalir didalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah medan putar (Soemarwanto,2006).

Metode *Pulse Width Modulation* (PWM) adalah salah satu metode yang dapat digunakan untuk mengatur kecepatan dari motor DC. Kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut. Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan (Alfarisi R.,2011). *Duty*

cycle adalah besarnya sinyal kontrol yang diberikan pada motor. Persamaan untuk perhitungan *duty cycle* ditunjukkan dalam persamaan 2.1 dengan T_{on} adalah periode logika tinggi, dan T adalah periode keseluruhan.

$$Duty\ cycle = T_{on}/T \times 100\% \quad (2.1)$$

Sinyal PWM secara umum ditunjukkan dalam Gambar 2.3.



Gambar 2.3 Sinyal PWM

Keterangan

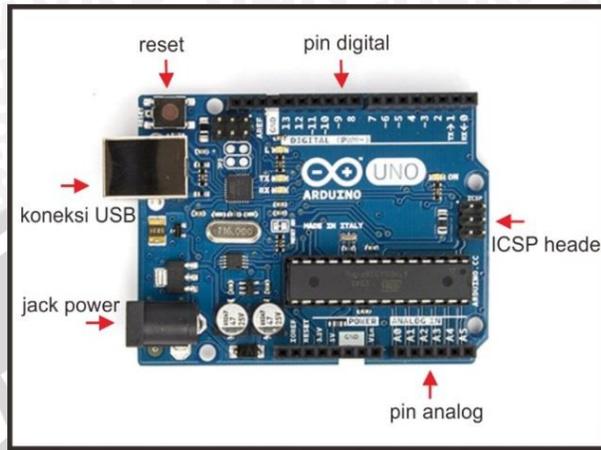
- T_{on} = Periode logika tinggi
- T = Periode keseluruhan
- V_o = Amplitudo

2.2 Arduino Uno

Arduino UNO adalah board mikrokontroler berbasis ATmega328. Memiliki 14 pin input dari output digital dimana 6 pin input tersebut dapat digunakan sebagai output PWM dan 6 pin input analog, 16 MHz osilator kristal, koneksi USB, jack power, ICSP header, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Board Arduino UNO ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang-ke adaptor-DC atau baterai untuk menjalankannya.

UNO berbeda dengan semua board sebelumnya dalam hal koneksi USB-to-serial yaitu menggunakan fitur Atmega8U2 yang diprogram sebagai konverter USB-to-serial berbeda dengan board sebelumnya yang menggunakan chip FTDI driver USB-to-serial.

Gambar mikrokontroler arduino UNO dan konfigurasi pin ATmega328 ditunjukkan dalam Gambar 2.4 dan Gambar 2.5.



Gambar 2.4 Arduino UNO

Sumber: arduino Datasheet

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT19/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLK0/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

Gambar 2.5 Konfigurasi Pin ATmega 328

Nama “UNO” berarti *satu* dalam bahasa Italia, untuk menandai peluncuran Arduino 1.0. UNO dan versi 1.0 akan menjadi versi referensi dari Arduino. UNO adalah yang terbaru dalam serangkaian board USB Arduino.

2.2.1 Catu Daya

Arduino UNO dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal (otomatis). 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 2.1mm konektor POWER. Ujung kepala dari baterai dapat dimasukkan kedalam Gnd dan Vin pin header dari konektor POWER. Kisaran kebutuhan daya yang disarankan untuk board UNO adalah 7 sampai dengan 12 volt, jika diberi daya kurang dari 7 volt kemungkinan pin 5v UNO dapat beroperasi tetapi tidak stabil kemudian jika diberi daya lebih dari 12V, regulator tegangan bisa panas dan dapat merusak board UNO.

2.2.2 Memori

ATmega328 memiliki 32 KB (dengan 0,5 KB digunakan untuk bootloader), 2 KB dari SRAM dan 1 KB EEPROM.

2.2.3 Input dan Output

Masing-masing dari 14 pin digital di UNO dapat digunakan sebagai input atau output, dengan menggunakan fungsi *pinMode ()*, *digitalWrite ()*, dan *digitalRead ()*, beroperasi dengan daya 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki internal pull-up resistor (secara default terputus) dari 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus:

a. Serial

0 (RX) dan 1 (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.

b. Eksternal menyela

2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interrupt pada nilai yang rendah, dengan batasan tepi naik atau turun, atau perubahan nilai. *PWM*: 3, 5, 6, 9, 10, dan 11. Menyediakan output PWM 8-bit dengan fungsi *analogWrite ()*.

c. SPI

10 (SS), 11 (Mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan *SPI library*.

d. LED

13. Ada built-in LED terhubung ke pin digital 13. Ketika pin bernilai nilai HIGH, LED on, ketika pin bernilai LOW, LED off.

UNO memiliki 6 masukan analog, berlabel A0 sampai dengan A5, yang masing-masing menyediakan 10 bit dengan resolusi (yaitu 1024 nilai yang berbeda). Selain itu, beberapa pin memiliki fungsi khusus, yaitu:

a. *I2C*

A4 (SDA) dan A5 (SCL). Dukungan I2C (TWI) komunikasi menggunakan perpustakaan *Wire*. Aref. Tegangan referensi (0 sampai 5V saja) untuk input analog. Digunakan dengan fungsi *analogReference* ().

b. *Reset*

Bawa baris ini LOW untuk me-reset mikrokontroler.

2.2.4 Komunikasi

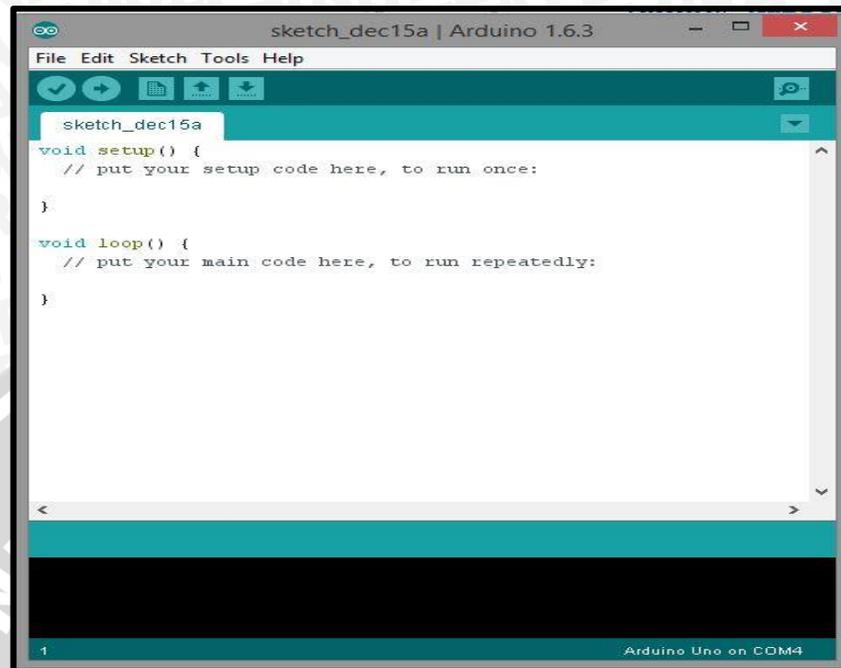
Arduino UNO memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, arduino lain, atau mikrokontroler lainnya. ATmega328 menyediakan UART TTL (5V) untuk komunikasi serial, yang tersedia di pin digital 0 (RX) dan 1 (TX). Sebuah ATmega8U2 sebagai saluran komunikasi serial melalui USB dan sebagai 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 atau dari board Arduino. LED RX dan TX di papan tulis akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dengan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah *SoftwareSerial library* memungkinkan untuk berkomunikasi secara serial pada salah satu pin digital pada board UNO. ATmega328 juga mendukung I2C (TWI) dan komunikasi SPI. Perangkat lunak Arduino termasuk perpustakaan *Kawat* untuk menyederhanakan penggunaan bus I2C.

2.2.5 Software

Pemrograman arduino dilakukan menggunakan *software* Arduino IDE. IDE Arduino terdiri dari:

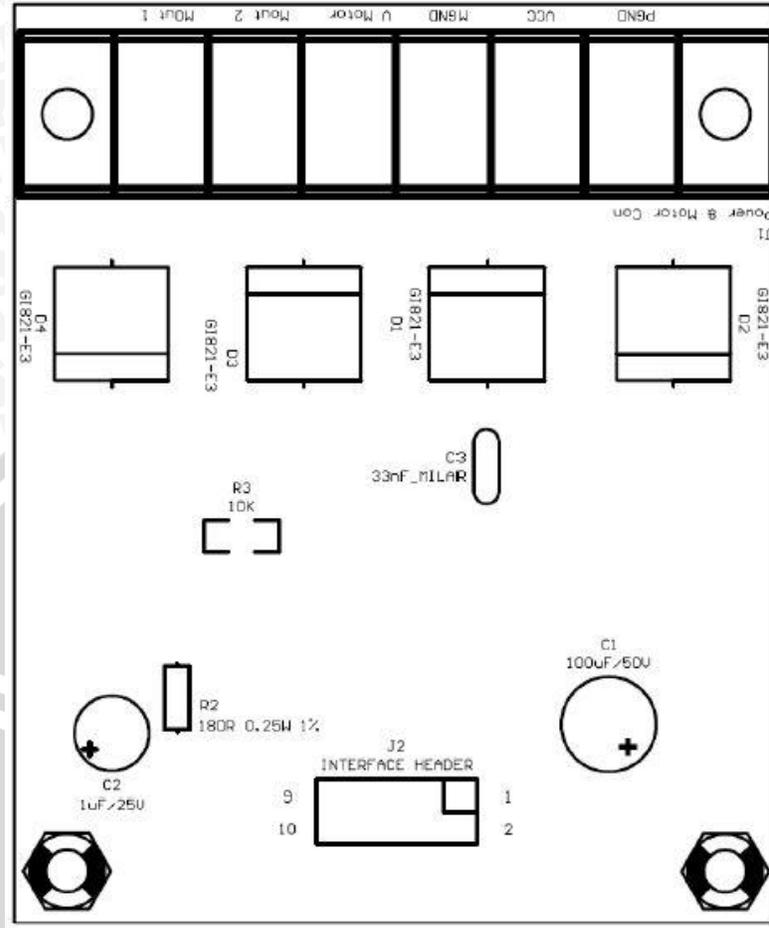
- a. *Editor* program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*.
- b. *Compiler*, sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner.
- c. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam *memory* di dalam papan Arduino.



Gambar 2.6 Software Arduino IDE

2.3 Driver EMS 5A H-Bridge

Embedded Module Series (EMS) 5A H-Bridge merupakan *driver* H-Bridge dengan arus kontinyu sampai dengan 5 A pada tegangan 5,5 Volt sampai 16 Volt. Modul ini dilengkapi dengan rangkaian sensor arus beban yang dapat digunakan sebagai umpan balik ke pengendali. Jalur catu daya input (VCC) terpisah dari jalur catu daya untuk beban (V mot). Dilengkapi dengan dioda eksternal untuk pengamanan beban induktif, proteksi hubungan singkat dan proteksi overtemperatur. Modul ini mampu men-*drive* beban-beban induktif seperti misalnya *relay*, solenoida, motor DC, motor *stepper*, dan berbagai macam beban lainnya. Gambar tata letak komponen EMS 5A H-Bridge ditunjukkan dalam Gambar 2.7.



Gambar 2.7 Tata Letak Komponen EMS 5 A H-Bridge
(Datasheet EMS 5A H-Bridge)

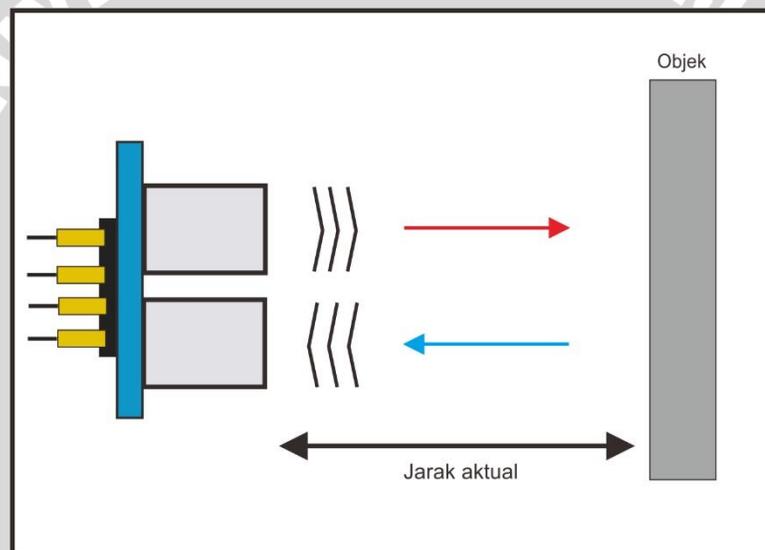
2.4 Sensor Ultrasonik HC-SR04

Sensor Ultrasonik merupakan sensor yang bekerja dengan cara memancarkan suatu gelombang dan kemudian menghitung waktu pantulan gelombang tersebut. Gelombang ultrasonic dapat merambat melalui zat padat, cair maupun gas. Sensor HC-SR04 memiliki jarak jangkauan minimal 2cm dan maksimal 400cm serta memiliki akurasi sebesar 3mm. Modul sensor Ultrasonic bekerja pada tegangan DC 5V, arus 15mA dan frekuensi 40Hz, dalam sebuah modul sensor ultrasonik terdiri dari pemancar, penerima gelombang ultrasonik dan unit kontrol. Gambar modul sensor *Ultrasonic* HC-SR04 ditunjukkan dalam Gambar 2.8.



Gambar 2.8 Sensor HC-SR04

Cara kerja sensor ultrasonik ditunjukkan dalam Gambar 2.9.



Gambar 2.9 Cara Kerja Sensor Ultrasonik

Sensor ultrasonik bekerja dengan mekanisme sebagai berikut:

1. Sinyal dipancarkan oleh pemancar ultrasonik dengan frekuensi tertentu dan dengan durasi waktu tertentu. Sinyal tersebut berfrekuensi diatas 20kHz. Untuk mengukur jarak benda (sensor jarak), frekuensi yang umum digunakan adalah 40kHz.
2. Sinyal yang dipancarkan akan merambat sebagai gelombang bunyi dengan kecepatan sekitar 340 m/s. Ketika menumbuk suatu benda, maka sinyal tersebut akan dipantulkan oleh benda tersebut.

3. Setelah gelombang pantulan sampai di alat penerima, maka sinyal tersebut akan diproses untuk menghitung jarak benda tersebut. Jarak benda dihitung berdasarkan rumus:

$$S = 340. t/2$$

dimana S merupakan jarak antara sensor ultrasonik dengan benda (bidang pantul), dan t adalah selisih antara waktu pemancaran gelombang oleh transmitter dan waktu ketika gelombang pantul diterima receiver (*Datasheet HC-SR04*).

2.5 Kontroler

Salah satu komponen sistem kontrol yang berfungsi mengolah sinyal umpan balik dan sinyal masukan acuan (*setpoint*) atau sinyal *error* mejadi sinyal kontrol. Sistem pengendalian dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengendalian adalah harus stabil. Di samping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif, yakni tolok ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikenai gangguan (Ogata K.,1997). Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam merespons *input*, dan bagaimana peredaman terhadap adanya lonjakan (*over shoot*).

Suatu sistem dikatakan stabil jika diberi gangguan maka sistem tersebut akan kembali ke keadaan *steady state* di mana *output* berada dalam keadaan tetap seperti tidak ada gangguan. Sistem dikatakan tidak stabil jika *outputnya* 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 respon transien berupa suatu osilasi teredam sebelum mencapai *steady state*.

2.5.1. Kontroler Proporsional (P)

Kontroler proporsional adalah sebuah kontroler yang memiliki karakteristik mempercepat respon. Hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ ditunjukkan dalam persamaan 2.2:

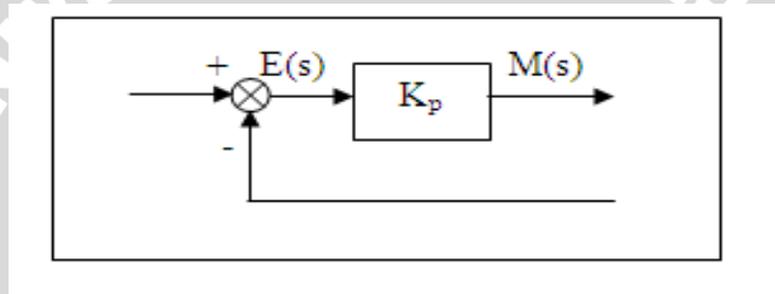
$$m(t)=K_p e(t) \tag{2.2}$$

atau, dalam besaran transformasi Laplace ditunjukkan dalam persamaan 2.3

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

dimana 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 2.10.



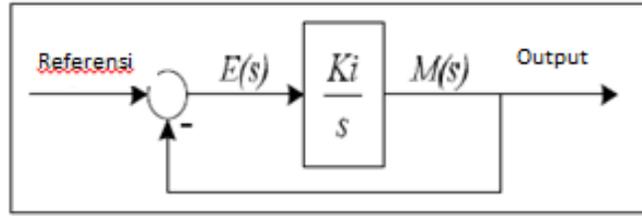
Gambar 2.10 Diagram Blok Kontroler Proporsional (Ogata K., 1997)

2.5.2. Kontroler Integral (I)

Kontroler integral memiliki kemampuan untuk mengurangi *offset* yang ditinggalkan oleh kontroler proporsional. Harga keluaran kontroler $m(t)$ diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak $e(t)$. Persamaan kontroler K_i ditunjukkan dalam persamaan 2.4.(Ogata K.,1997).

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

K_i adalah penguatan integral. Gambar 2.11 menunjukkan diagram blok kontroler integral.



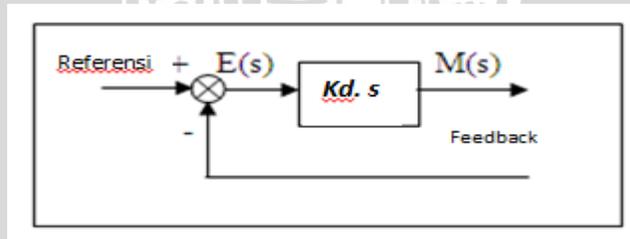
Gambar 2.11 Diagram Blok Kontroler Integral (Ogata., 1997)

2.5.3. Kontroler Differensial (D)

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 differensial 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 $e(t)$ dinyatakan dalam persamaan 2.5

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

Kd adalah penguatan differensial. Gambar 2.12 menunjukkan diagram blok kontroler differensial.



Gambar 2.12 Diagram Blok Kontroler Differensial (Ogata K., 1997)

2.5.4. Kontroler Proporsional Integral Differensial (PID)

Gabungan aksi kontrol proporsional, integral, dan differensial mempunyai keunggulan dibandingkan dengan masing-masing dari tiga aksi kontrol tersebut. Masing-masing kontroler P, I, maupun D berfungsi untuk mempercepat reaksi

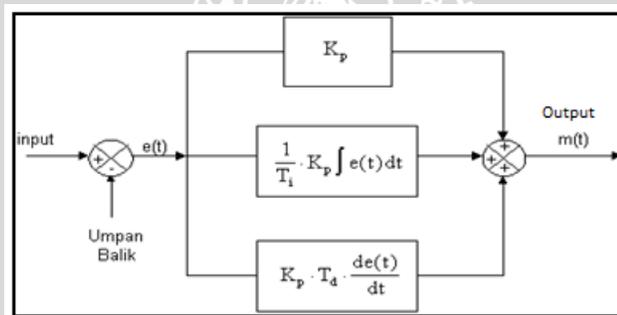
sistem, menghilangkan *offset*, dan mendapatkan energi ekstra ketika terjadi perubahan *load*. Persamaan kontroler PID ini dapat dinyatakan dalam persamaan 2.6

$$m(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \cdot e(t)dt + K_p \cdot T_d \frac{de(t)}{dt} \quad (2.6)$$

Dalam transformasi Laplace dinyatakan dalam persamaan 2.7

$$\frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \quad (2.7)$$

T_i adalah waktu integral dan T_d adalah waktu derivatif. Gambar 2.13 menunjukkan diagram blok kontroler PID.

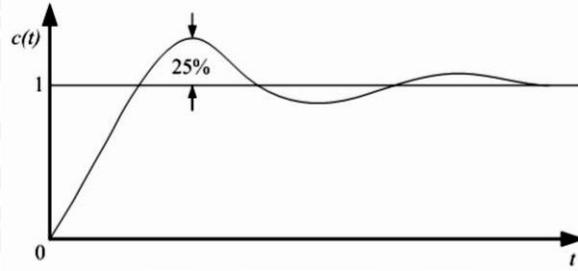


Gambar 2.13 Diagram Blok Kontroler PID (Ogata K., 1997)

2.6 Ziegler-Nichols

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

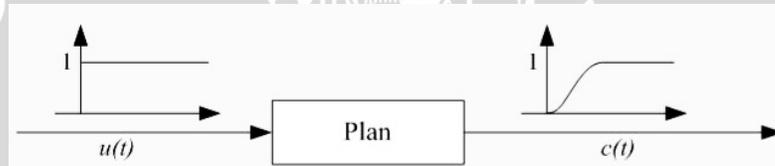
Terdapat dua metode yang disebut dengan aturan penalaan Ziegler-Nichols, pada kedua metode tersebut memiliki tujuan yang sama yaitu untuk mencapai 25% *maximum overshoot* pada respon unit step, seperti ditunjukkan dalam Gambar 2.14.



Gambar 2.14. Kurva Respon Unit Step yang Menunjukkan 25% Maximum Overshoot
(Ogatta, K.,1997)

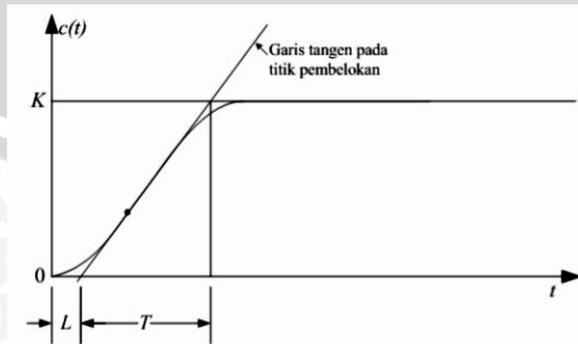
2.6.1 Metode Pertama

Metode pertama atau sering disebut metode kurva reaksi, respon dari *plant* dapat dapat diperoleh secara eksperimental dengan masukan berupa unit step, seperti yang ditunjukkan dalam Gambar 2.15



Gambar 2.15 Respon Plant Terhadap Masukan Berupa Unit Step
(Ogata, K.,1997)

Jika dalam *plant* tersebut terdapat integrator atau *dominan complex-conjugate poles*, maka kurva respon unit step berbentuk seperti huruf S, seperti dalam Gambar 2.16 jika respon tidak memberikan bentuk kurva S, maka metode ini tidak berlaku.(Ogata, K., 1997).



Gambar 2.16 Kurva Respon yang Berbentuk S (Ogata, K., 1997)

Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda L dan konstanta waktu T . Waktu tunda dan konstanta waktu ditentukan dengan menggambar sebuah garis tangen pada titik pembelokan dari kurva S, dan menentukan perpotongan antara garis tangen dengan sumbu waktu t dan sumbu $c(t) = K$, seperti yang telah ditunjukkan dalam Gambar 2.12 Fungsi alih $C(s)/U(s)$ dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut (Ogata, K., 1997)

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts+1} \quad (2.14)$$

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

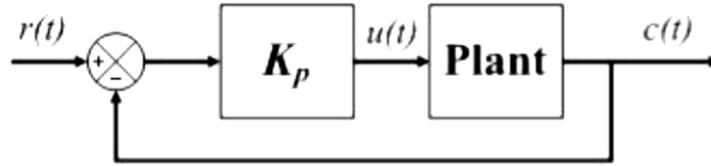
Tabel 2.1 Aturan Penalaan Ziegler-Nichols Berdasarkan Respon Unit Step Dari Plan

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

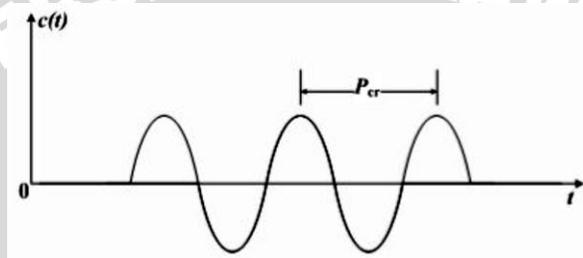
2.6.2 Metode Kedua

Dalam metode kedua Ziegler-Nichols, mula-mula yang dilakukan adalah membuat $T_i = \infty$ dan $T_d = 0$. Kemudian hanya dengan menggunakan tindakan kontrol proporsional, harga ditingkatkan dari nol ke suatu nilai kritis K_{cr} , disini mula-mula keluaran memiliki osilasi yang berkesinambungan (Jika keluaran tidak memiliki osilasi berkesinambungan untuk nilai K_p manapun yang telah diambil, maka metode ini tidak berlaku). Dari keluaran yang berosilasi secara berkesinambungan, penguatan kritis K_{cr} dan periode P_{cr} dapat ditentukan. Diagram blok sistem loop tertutup dengan kontroler proporsional dapat dilihat dalam Gambar 2.17. dan untuk osilasi berkesinambungan dengan periode P_{cr} dapat dilihat dalam gambar 2.18. Ziegler dan Nichols menyarankan

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



Gambar 2.17. Sistem Loop Tertutup dengan Kontroler Proporsional (Ogata, K., 1997)



Gambar 2.18. Osilasi Berkesinambungan dengan Periode P_{cr} (Ogata, K., 1997)

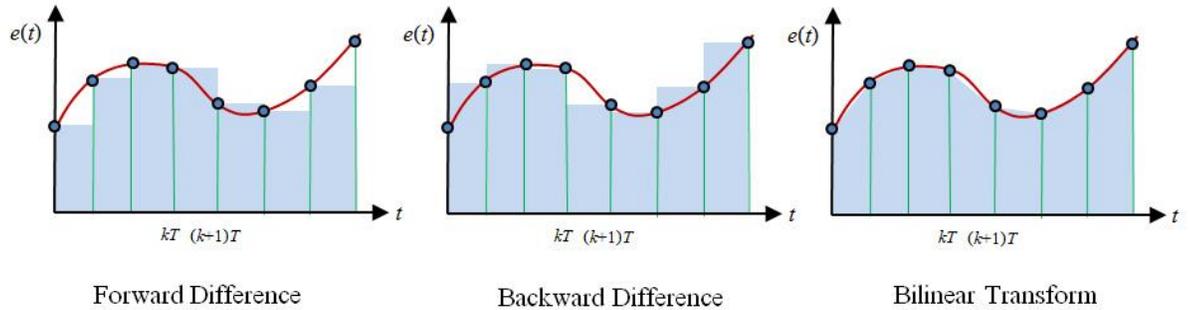
Tabel 2.2 Aturan Dasar Ziegler-Nichols Berdasarkan Critical Gain K_{cr} dan Critical Period P_{cr} (Ogata, k.,1997)

Type Kontroler	K_p	T_i	T_d
P	$0.5 K_{cr}$	∞	0
PI	$0.45 K_{cr}$	$\frac{1}{1.2} P_{cr}$	0
PID	$0.60 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

2.7 Diskritisasi

Diskritisasi dilakukan karena persamaan kontroler PID yang didapat dari transformasi laplace masih dalam bentuk kontinyu. Banyak cara yang dapat digunakan untuk proses diskritisasi (mengubah bentuk analog menjadi diskrit), tiga diantaranya adalah *forward difference*, *backward difference*, dan *bilinear transform*. Semua metode yang digunakan hanya merupakan pendekatan (*approximation*), sehingga hasilnya tidak akan sama persis dengan bentuk analog. Hal ini dikarenakan bentuk diskrit selalu ada sebagian informasi yang hilang, yaitu informasi yang terjadi antarasatu cuplikan dengan

cuplikan lainnya walaupun frekuensi cuplik yang digunakan tinggi dan karena formula matematika yang digunakan dalam proses diskritisasi juga diturunkan berdasarkan pendekatan agar lebih mudah digunakan. 3 metode diskritisasi ditunjukkan dalam Gambar 2.19 .



Gambar 2.19 Metode Konversi

Proses diskritisasi dilakukan dengan mengganti operator s pada transformasi Laplace menjadi bentuk transformasi Z menggunakan salah satu dari tiga metode diatas yaitu:

Backward difference

$$s = \frac{1-z^{-1}}{Ts}$$

Forward difference

$$s = \frac{1-z^{-1}}{Ts z^{-1}}$$

Bilinear transform

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

Algoritma sudah dalam bentuk diskrit yang dinyatakan dalam transformasi Z. Untuk implementasinya, bentuk transformasi Z perlu diubah menjadi *time domain*, dengan mengganti operator z^0 menjadi waktu saat ini (k), dan z^{-1} menjadi waktu sesaat sebelumnya atau satu kali waktu tunda ($k - 1$) (Wain Y, 2009).



BAB III

METODE PENELITIAN

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

3.1. Penentuan Spesifikasi Alat

Adapun spesifikasi alat sebagai berikut:

1. Tangki pengisian menggunakan *Plant Filling Tank (73624)*
2. Pompa motor DC 12v sebagai aktuator dari *Plant*
3. Sensor Ultrasonik HC-SR04
4. Kontroler yang digunakan adalah kontroler Proporsional, Integral, dan Diferensial (PID)
5. Perangkat kontrol yang digunakan untuk akuisisi data adalah Arduino UNO

3.2. Perancangan Sistem

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik dan pemrogramannya. Hal ini dilakukan agar proses identifikasi sistem dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan. Perancangan sistem yang dilakukan meliputi:

3.2.1 Perancangan Blok Diagram Sistem

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

3.2.2 Perancangan Perangkat Keras

Perancangan perangkat keras dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya. Hal ini dimaksudkan agar sistem pengendali ketinggian air dapat berjalan sesuai dengan rencana

yang telah ditetapkan. Perancangan perangkat keras yang dilakukan meliputi:

- a. Pembuatan modul rangkaian
- b. Penentuan rangkaian elektrik meliputi:
 1. Rangkaian catu daya
 2. Rangkaian driver EMS 5A H-Bridge
 3. Rangkaian sensor ultrasonic HC-SR04
 4. Rangkaian mikrokontroler Arduino

3.3. Pengujian Sistem

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

3.4. Pengambilan Kesimpulan

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

BAB IV PERANCANGAN DAN PEMBUATAN ALAT

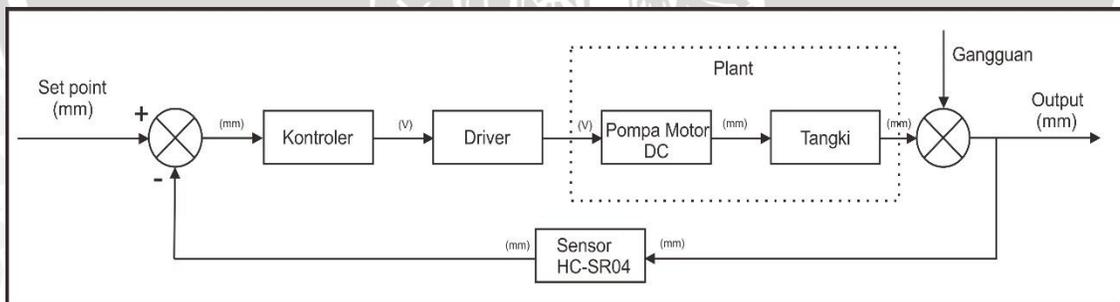
4.1 Perancangan Sistem

Perancangan dan pembuatan alat terdiri dari dua bagian, yaitu perancangan dan pembuatan perangkat keras serta perancangan dan pembuatan perangkat lunak. Perancangan dan pembuatan alat dilakukan secara bertahap untuk memudahkan analisis sistem. Beberapa aspek yang perlu dijelaskan dalam bab ini meliputi:

1. Cara kerja alat
2. Spesifikasi alat
3. Perancangan perangkat keras (rangkaiian mikrokontroler, rangkaian driver motor, dan pompa)
4. Perancangan perangkat lunak (perancangan kontrol PID)

4.2 Diagram Blok Sistem

Diagram blok sistem yang dirancang ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Diagram Blok Sistem

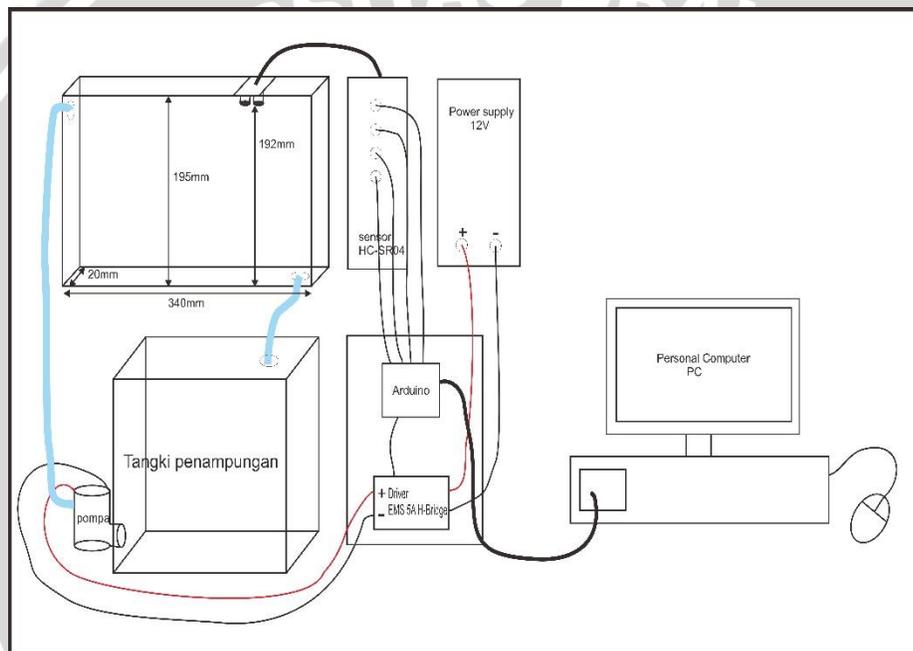
Keterangan dari blok diagram sistem tersebut adalah sebagai berikut:

1. *Set point* sistem berupa ketinggian air (mm) dalam tangki pengisian sebesar yang diinginkan
2. Pusat pengendalian sistem menggunakan mikrokontroler Arduino yang memberikan sinyal keluaran berupa *pulse width modulation* (PWM) ke *driver* EMS 5A H-Bridge

3. *Plant* sistem berupa ketinggian air di dalam tangki dan pompa motor DC yang sekaligus sebagai actuator
4. Sebagai *feed back* digunakan sensor ultrasonic HC-SR04 yang dihubungkan dengan mikrokontroler Arduino pada pin digital
5. Gangguan pada *plant* berupa pemasangan sekat-sekat berlubang di tangki pengisian
6. Keluaran dari sistem merupakan ketinggian air (mm) pada tangki pengisian

4.3 Spesifikasi Alat

Skema perancangan sistem dapat dilihat dalam Gambar 4.2.



Gambar 4.2 Skema Perancangan

Spesifikasi sistem pengendali ketinggian air menggunakan PID digital adalah sebagai berikut:

1. Rentang ketinggian air dalam tangki pengisian sebesar 19,2cm
2. Dimensi tangki sebesar panjang 34cm, lebar 2cm, dan tinggi 19,5cm
3. Sensor yang digunakan adalah sensor ultrasonic HC-SR04 yang memiliki rentang pengukuran dari 2cm sampai dengan 400cm
4. Aktuator menggunakan pompa motor DC yang mendapat sinyal masukan dari *output* PWM Driver EMS 5A H-Bridge

5. Catu daya yang digunakan adalah sebesar 12 Volt DC.
6. Penampil data hasil pengontrolan menggunakan perangkat komputer melalui program *serial monitor* Arduino IDE.

4.4 Prinsip Kerja Sistem

Cara kerja sistem pengendali ketinggian air pada *plant* 73624 adalah sebagai berikut:

1. Catu daya yang digunakan terdiri atas dua macam, yaitu:
 - a. 5 VDC yang digunakan sebagai catu daya bagi mikrokontroler Arduino
 - b. 12 VDC yang digunakan sebagai catu daya pompa motor DC
2. Menggunakan sensor ultrasonic HC-SR04 dimana sinyal keluaran dari sensor sudah berupa data digital sehingga dapat dikoneksikan langsung sebagai masukan pada mikrokontroler Arduino kemudian diproses menjadi sinyal kontrol berupa PWM.
3. Sinyal kontrol dari mikrokontroler arduino difungsikan sebagai masukan ke *driver* EMS 5A H-Bridge. *Driver* berfungsi menguatkan sinyal yang dihasilkan mikrokontroler Arduino dari 0-5 Volt menjadi 5-12 Volt.
4. Pompa motor DC akan mengatur kecepatan ketika pembacaan ketinggian air oleh sensor ultrasonic HC-SR04 belum mencapai *set point*.
5. Kecepatan pengisian oleh pompa motor DC akan berubah sesuai dengan pembacaan ketinggian oleh sensor.
6. Gangguan diberikan di dalam *plant* yang berupa sekat-sekat berlubang pada saat ketinggian air telah mencapai *set point* agar aliran air di dalam tangki dapat bervariasi dan dapat melihat aksi kontrol yang dilakukan oleh kontroler.
7. Data hasil pengontrolan ditampilkan pada program *serial monitor* Arduino IDE berupa data *Output* PWM, hasil perhitungan kontroler, dan jarak.

4.5 Perancangan Perangkat Keras

Berdasarkan diagram blok perancangan alat yang telah disusun, perancangan perangkat keras meliputi perancangan *plant* pengendali ketinggian air serta perancangan listrik yang meliputi rangkaian *driver* EMS 5A H-Bridge, dan modul *input output* mikrokontroler Arduino.

Interface Header pada driver EMS H-Bridge 5 A berfungsi sebagai input untuk antarmuka dengan input-output digital serta output analog dari modul H-Bridge. Berikut deskripsi dari masing-masing pin ditunjukkan dalam Tabel 4.1.

Tabel 4.1 Fungsi Pin EMS H-Bridge 5 A

No. Pin	Fungsi
1	Pin <i>input</i> untuk menentukan <i>output</i> MOUT1
2	Pin <i>input</i> untuk menentukan <i>output</i> MOUT2
3	Pin <i>enable</i> untuk <i>output</i> MOUT1 Diberi logika <i>high</i> untuk mengaktifkan <i>half</i> H-Bridge 1, diberi logika <i>low</i> secara eksternal untuk menonaktifkan <i>half</i> H-Bridge 1. Jika terjadi kondisi <i>fault</i> (<i>thermal shutdown</i> , <i>undervoltage</i> , <i>overvoltage</i> , dan sebagainya), maka pin ini akan ditarik <i>low</i> secara internal oleh modul H-Bridge untuk melaporkan adanya kondisi <i>fault</i>
4	Pin <i>enable</i> untuk <i>output</i> MOUT2 Diberi logika <i>high</i> untuk mengaktifkan <i>half</i> H-Bridge 2, diberi logika <i>low</i> secara eksternal untuk menonaktifkan <i>half</i> H-Bridge 2. Jika terjadi kondisi <i>fault</i> (<i>thermal shutdown</i> , <i>undervoltage</i> , <i>overvoltage</i> , dan sebagainya), maka pin ini akan ditarik <i>low</i> secara internal oleh modul H-Bridge untuk melaporkan adanya kondisi <i>fault</i>
5	<i>Output</i> tegangan analog yang berbanding lurus dengan arus beban (<i>range output</i> 0 – 5 Volt)
6	Pin <i>input</i> untuk mengatur kerja modul H-Bridge secara PWM
7, 9	Terhubung ke catu daya untuk <i>input</i> (5 Volt)
8, 10	Titik referensi untuk catu daya <i>input</i>

4.5.2 Konfigurasi I/O Arduino

Mikrokontroler Arduino berfungsi sebagai pusat pengolah data utama dalam proses pengendalian. Konfigurasi I/O pada mikrokontroler Arduino ditunjukkan dalam Tabel 4.2.

Tabel 4.2 Fungsi Pin Arduino

No.	Pin	Fungsi
1	D9	Digunakan untuk PWM driver
2	D10	Digunakan untuk TRIG sensor
3	D11	Digunakan untuk ECHO sensor

4.6 Perancangan Kontroler

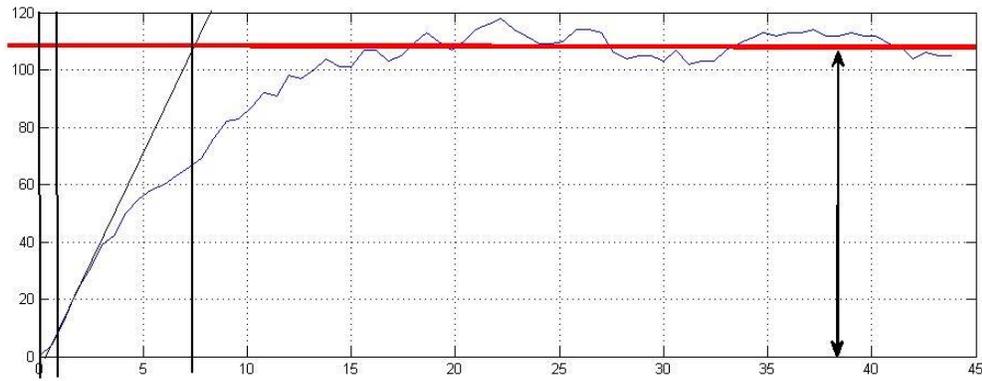
Untuk memenuhi tujuan performansi *loop* yang diinginkan, maka perlu ditambahkan kontroler pada sistem tersebut. Kontroler yang dipilih adalah kontroler proporsional integral diferensial (PID). Parameter kontrol PID didapatkan dengan menggunakan metode *Ziegler-Nichols*, dimana *plant* dirancang secara *open loop* dan diberi tegangan masukan ke *plant*, kemudian diamati responnya.

Sebelum dilakukan perancangan kontroler perlu diketahui terlebih dahulu bagaimana karakteristik *plant*. Hal ini didapatkan dengan melakukan pengujian karakteristik *plant* terlebih dahulu dengan cara diberi masukan unit *step*. Dari hasil pengujian didapatkan bahwa *plant* dapat bekerja sampai dengan PWM 80. Ketika diberikan PWM lebih dari 80 pompa terus bekerja sampai air melebihi batas penampungan tangki pengisian. Data hasil pengujian karakteristik *plant* dapat dilihat dalam Tabel 4.3.

Tabel 4.3 Hasil Pengujian Karakteristik Plant Tangki Air (73426)

PWM	Ketinggian (mm)
0	0
10	0
20	3
30	5
40	23
50	50
60	75
70	115
80	167
90	>195
255	>195

Hasil pencarian karakteristik *plant* menunjukkan bahwa *plant* dapat bekerja sampai dengan PWM 80. Untuk perancangan kontroler diberikan PWM masukan sebesar 70 dan didapatkan hasil sebagai berikut (Gambar 4.5):

Gambar 4.5 Grafik *Output Plant*

Dari grafik *output plant* pada gambar 4.5 didapatkan nilai:

$$\Delta u = \text{PWM } 70 = 4,2 \text{ V}$$

$$\Delta y = 115 \text{ mm}$$

$$L = 1,8 \text{ detik}$$

$$T = 5,2 \text{ detik}$$

Maka,

$$K = \frac{\Delta y}{\Delta u} = \frac{115}{4,2} = 27,38 \text{ mm/V}$$

Setelah mendapatkan parameter yang dibutuhkan maka kita bisa mendapatkan fungsi alih *plant* seperti yang ditunjukkan dalam persamaan

$$G(s) = \frac{K e^{-Ls}}{T_s + 1} = \frac{27,38 e^{-1,8}}{5,2 s + 1}$$

Dan diperoleh nilai K_p , K_i , K_d sebagai berikut:

$$K_p = 1.2 \times \frac{T}{L} = 1.2 \times \frac{5,2}{1,8} = 3,46$$

$$K_i = \frac{K_p}{T_i} = \frac{3,46}{2 \times L} = \frac{3,46}{2 \times 1,8} = 0,96$$

$$K_d = K_p \times T_d = 3,46 \times (0.5 \times 1,8) = 3,46 \times 0.9 = 3,1$$

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

$$C(s) = \left(K_p + \frac{K_i}{s} + K_d s \right) e(s) \quad (4.1)$$

Persamaan tersebut masih dalam bentuk kontinyu dan belum bisa dimasukkan ke mikrokontroler maka harus diubah menjadi bentuk diskrit melalui transformasi Z dengan cara mengganti operator s pada transformasi *Laplace*. Digunakan metode *Backward Difference* sehingga nilai operator s pada Laplace setara dengan:

$$s = \frac{1-z^{-1}}{T_s} \quad (4.2)$$

Dimana T_s : Periode sampling

Jika persamaan (4.2) disubstitusikan ke dalam persamaan (4.1) menjadi:

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

$$\text{Kontroler Proporsional} : C_p(z) = K_p E(z)$$

$$\text{Kontroler Integral} : C_i(z) = \frac{K_i T_s}{1-z^{-1}} \times E(z)$$

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

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

$$\text{Kontroler Diferensial} : C_d(z) = \frac{K_d}{T_s} (1-z^{-1}) \times E(z)$$

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

Dari persamaan di atas kemudian diubah kedalam persamaan beda sehingga didapatkan persamaan berikut:

- Kontroler Proporsional : $C_p(k) = K_p \times E(k)$
- Kontroler Integral : $C_i(k) = C_i(k - 1) + K_i T_s \times E(k)$
- Kontroler Diferensial : $\frac{K_d}{T_s} [E(k) - E(k - 1)]$

Jika digabungkan menjadi:

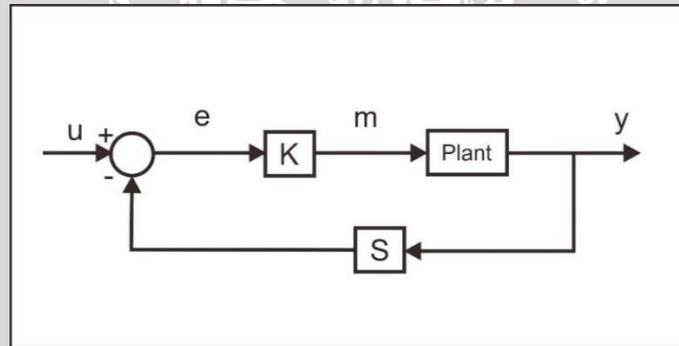
$$C(k) = C_p(k) + C_i(k) + C_d(k)$$

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

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

4.6.1 Performansi Sistem (Perancangan)

Untuk mengetahui kinerja dari sistem perancangan dilakukan perhitungan performansi sistem diantaranya *time peak (tp)*, maksimal *overshoot (mp)*, *time settling (ts)*, *time rise (tr)*, *time delay (td)*, dan *error steady state*.



Gambar 4.6 Blok Diagram Sistem

$$\frac{M(s)}{E(s)} = \frac{Kp_s + Ki + Kd_s^2}{s}$$

$$\frac{Y(s)}{M(s)} = \frac{K}{Ts + 1}$$

$$\frac{Y(s)}{E(s)} = \frac{K(Kp_s + Ki + Kd_s^2)}{s(Ts + 1)}$$

$$\frac{Y(s)}{U(s)} = \frac{\frac{K(Kp_s + Ki + Kd_{s^2})}{s(Ts + 1)}}{1 + \frac{K(Kp_s + Ki + Kd_{s^2})}{s(Ts + 1)}} \cdot K$$

$$\text{Fungsi Alih Plant} = \frac{27,38e^{-1,8}}{5,2s+1}$$

$$\text{Fungsi Alih Kontroler} = \frac{3,46s+0,96+3,1s^2}{s}$$

$$\text{Fungsi Alih Sensor } K = \frac{\text{span out}}{\text{span in}} = \frac{15-0}{40-0} = 0,325$$

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

$$\frac{Y(s)}{E(s)} = \frac{27,38e^{-1,8}(3,46s + 0,96 + 3,1s^2)}{s(5,2s + 1)}$$

$$\frac{Y(s)}{E(s)} = \frac{4,52(3,46s + 0,96 + 3,1s^2)}{s(5,2s + 1)}$$

$$\frac{Y(s)}{E(s)} = \frac{15,65s + 4,43 + 14,03s^2}{5,2s^2 + s}$$

$$\frac{Y(s)}{U(s)} = \frac{\frac{15,65s + 4,43 + 14,03s^2}{5,2s^2 + s}}{1 + \frac{15,65s + 4,43 + 14,03s^2}{5,2s^2 + s}} \cdot 0,325$$

$$\frac{Y(s)}{U(s)} = \frac{\frac{15,65s + 4,43 + 14,03s^2}{5,2s^2 + s}}{1 + \frac{5,08s + 1,43 + 4,55s^2}{5,2s^2 + s}}$$

$$\frac{Y(s)}{U(s)} = \frac{\frac{15,65s + 4,43 + 14,03s^2}{5,2s^2 + s}}{\frac{6,08s + 1,43 + 9,75s^2}{5,2s^2 + s}}$$

$$\frac{Y(s)}{U(s)} = \frac{15,65s + 4,43 + 14,03s^2}{6,08s + 1,43 + 9,75s^2} \cdot 5,2s^2 + s$$

$$\omega_n^2 = 1,43$$

$$\omega_n = \sqrt{1,43} = 1,19$$

$$2\xi\omega_n = 6,08$$

$$\xi\omega_n = 3,04$$

$$\xi = \frac{3,04}{1,19} = 2,55$$

$$\omega_d = \omega_n \sqrt{1 - \xi^2}$$

$$\omega_d = 1,19 \cdot 2,3 = 2,73$$

$$\sigma = \xi \omega_n$$

$$\sigma = 2,55 \cdot 1,19 = 3,03$$

$$\beta = \tan^{-1} \frac{\omega_d}{\sigma}$$

$$\beta = 0,73$$

Performansi Sistem Perancangan

$$1. \text{ Time peak (tp)} = \frac{\pi}{\omega_n \sqrt{1 - \xi^2}}$$

$$= \frac{3,14}{1,19 \cdot 2,3} = 1,15$$

$$2. \text{ Maksimal overshoot} = 100 \cdot e^{-\left(\frac{\sigma}{\omega_d}\right)\pi} \%$$

$$= 100 \cdot e^{-3,48} = 3,08 \%$$

$$3. \text{ Time settling (ts)} = \frac{4}{\xi \omega_n}$$

$$= \frac{4}{3,04} = 1,31$$

$$4. \text{ Time rise (tr)} = \frac{\pi - \beta}{\omega_d}$$

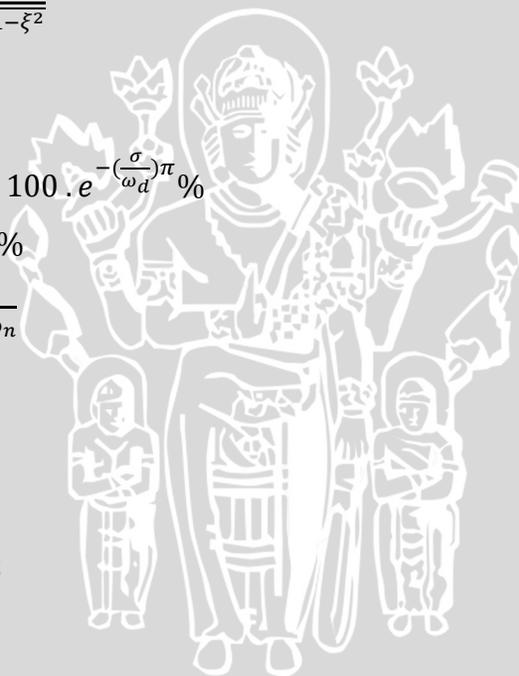
$$= \frac{3,14 - 0,73}{2,73} = 0,88$$

$$5. \text{ Time delay (td)} = \frac{1}{2} tr$$

$$= \frac{1}{2} \cdot 0,88 = 0,44$$

$$6. \text{ Error steady state} = \frac{2\xi}{\omega_n}$$

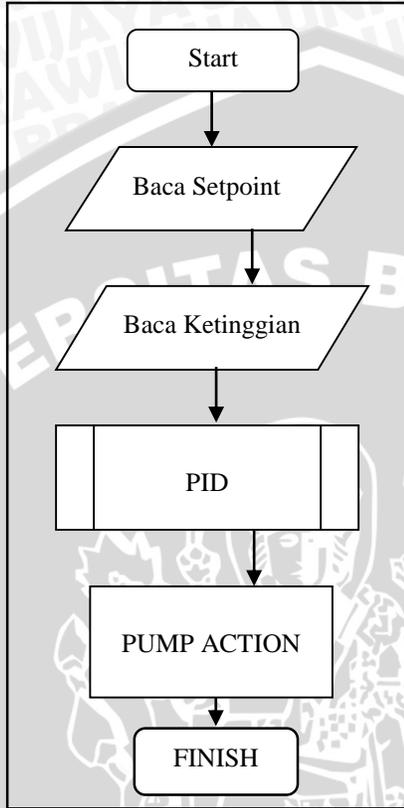
$$= \frac{2 \cdot 2,55}{1,19} = 4,28$$



4.7 Perancangan Perangkat Lunak

4.7.1 Flowchart Keseluruhan Sistem

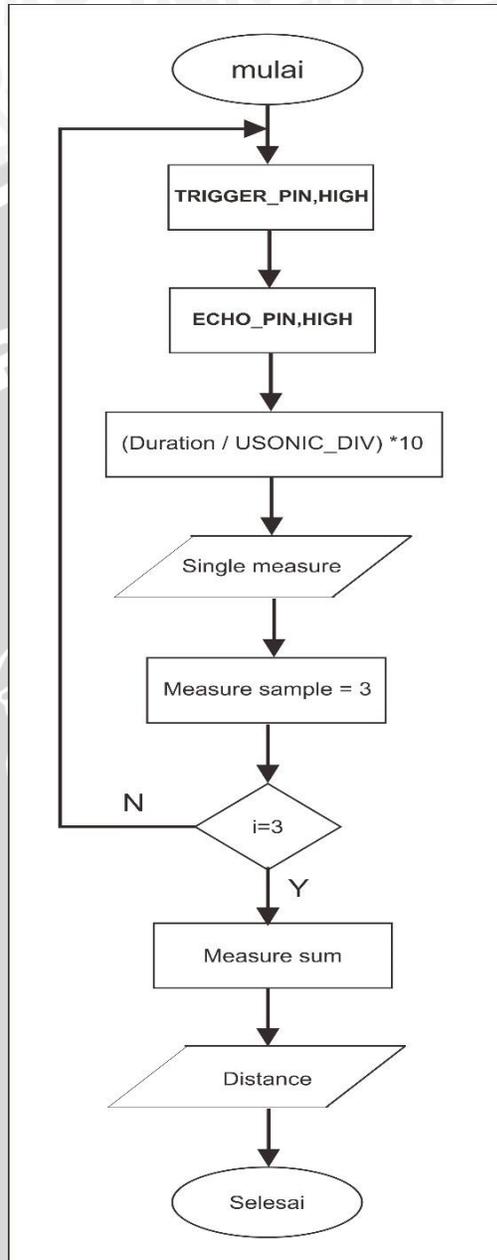
Flowchart keseluruhan sistem ditunjukkan pada Gambar 4.7



Gambar 4.7 Flowchart Keseluruhan Sistem

4.7.2 Flowchart Pembacaan Sensor

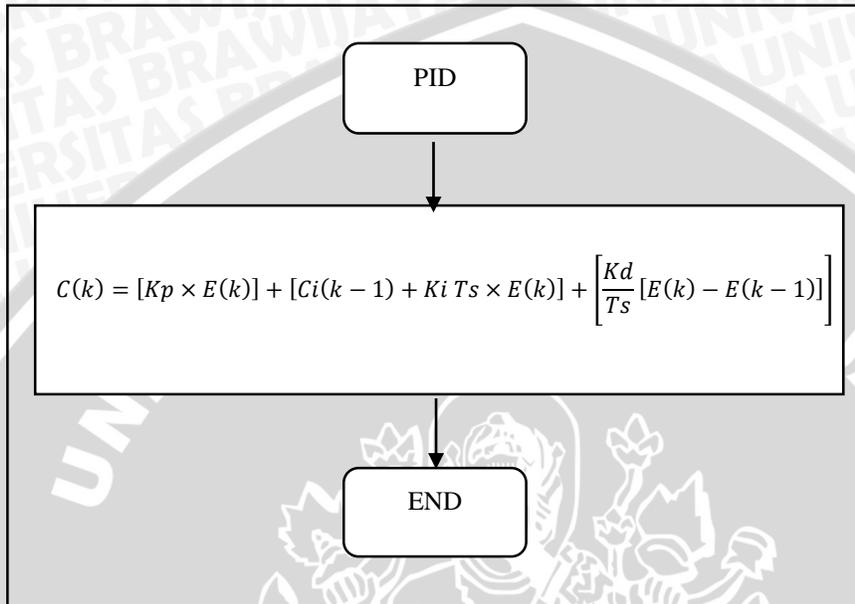
Flowchart keseluruhan sistem ditunjukkan pada Gambar 4.8



Gambar 4.8 Flowchart Pembacaan Sensor

4.7.3 Flowchart Kontrol PID

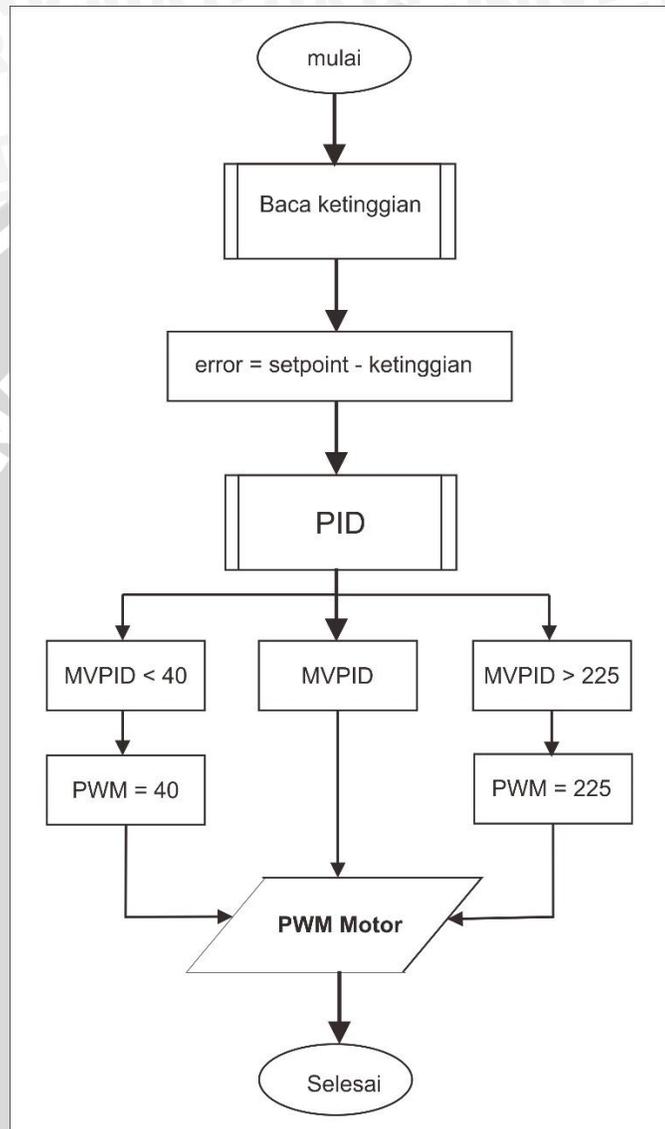
Flowchart kontrol PID ditunjukkan pada Gambar 4.9



Gambar 4.9 Flowchart Kontrol PID

4.7.4 Flowchart pump action

Flowchart pump action ditunjukkan pada gambar 4.10



Gambar 4.10 Flowchart Pump Action



BAB V

PENGUJIAN DAN ANALISIS

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

5.1 Pengujian Sensor Ultrasonik HC-SR04

a. Tujuan

Mengetahui kemampuan sensor ultrasonik HC-SR04 dalam membaca jarak

b. Peralatan yang digunakan

- Sensor ultrasonik HC-SR04
- Arduino Uno
- Kabel *jumper*
- Penggaris
- Perangkat komputer

c. Langkah pengujian

1. Menghubungkan sensor HC-SR04 ke arduino menggunakan kabel *jumper*
2. Meletakkan sensor HC-SR04 pada tempat sensor di tangki pengisian
3. Menutup lubang keluaran tangki pengisian dan memberi air secara bertahap sesuai dengan ketinggian yang diinginkan
4. Mencatat data ketinggian keluaran dari serial monitor arduino untuk melihat kelinierannya

d. Hasil Pengujian

Data hasil pengujian sensor ultrasonik HC-SR04 ditunjukkan dalam Tabel 5.1

Tabel 5.1 Hasil Pengujian Sensor Ultrasonik HC-SR04

Jarak (cm)	Jarak sensor (mm)	Jarak (cm)	Jarak sensor (mm)
1	9	11	113
2	22	12	121
3	28	13	131
4	46	14	139
5	52	15	148
6	59	16	161
7	72	17	169
8	80	18	178
9	93	19	191
10	98		

Perhitungan kesalahan pembacaan:

$$\% \text{ kesalahan} = \frac{(\sum \text{Hasil Pengukuran} - \sum \text{Referensi})}{\sum \text{Referensi}} \times 100\%$$

$$\% \text{ kesalahan} = \frac{(1910 - 1900)}{1900} \times 100\% = 0.52\%$$

Dari hasil perhitungan di atas dapat dilihat bahwa hasil pembacaan sensor ultrasonik HC-SR04 memiliki kesalahan pembacaan yang kecil sehingga sensor dapat digunakan.

5.2 Pengujian Driver EMS H-Bridge 5A

a. Tujuan

Pengujian ini bertujuan untuk mengetahui karakteristik *driver* motor terhadap perubahan *duty cycle*, karena nilai tegangan keluaran dari *driver* akan mempengaruhi kecepatan motor DC.

b. Peralatan yang digunakan

- *Driver* EMS H-Bridge 5A
- *Digital Oscilloscope* Vellemen PCSU1000
- Arduino Uno

c. Langkah pengujian

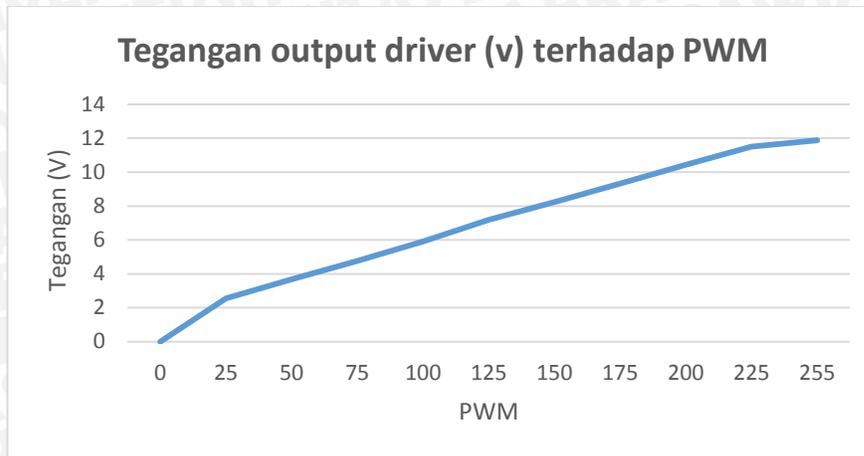
1. Menghubungkan masukan *driver* EMS H-Bridge 5A ke Arduino Uno
2. Menghubungkan keluaran *driver* EMS H-Bridge 5A ke *Digital Oscilloscope* Vellemen PCSU1000 dan voltmeter
3. Arduino Uno memberikan instruksi sinyal masukan pada pin PWM *driver* motor EMS H-Bridge 5A dengan PWM 0-255
4. Mencatat hasil perbandingan antara tegangan *output driver* dengan PWM.

d. Hasil pengujian

Besar perbandingan antara tegangan *output driver* dengan PWM dapat dilihat dalam Tabel 5.2 dan Gambar 5.1.

Tabel 5.2 Hasil Pengujian *Driver* EMS H-Bridge 5A

No	PWM	Duty Cycle (%)	Tegangan (V)
1	0	?	0
2	25	9,8	2,54
3	50	19,6	3,68
4	75	29,8	4,77
5	100	39,6	5,9
6	125	49	7,1
7	150	58,8	8.23
8	175	68,6	9,32
9	200	78,4	10,44
10	225	88,3	11,5
11	255	?	11,87



Gambar 5.1 Grafik Perbandingan Tegangan Output Driver terhadap PWM

Berdasarkan Gambar 5.1 dapat disimpulkan bahwa *driver* EMS H-Bridge 5A memiliki tingkat linieritas yang baik pada rentang PWM 25 sampai 225 dan *driver* EMS H-Bridge 5A baik untuk digunakan.

5.3 Pengujian Keseluruhan Sistem

a. Tujuan

Pengujian sistem secara keseluruhan dilakukan untuk mengetahui seberapa besar nilai kontroler yang dibutuhkan agar sistem bekerja sesuai dengan *set point*, yaitu 7 - 9cm dengan kenaikan 1cm.

b. Peralatan yang digunakan

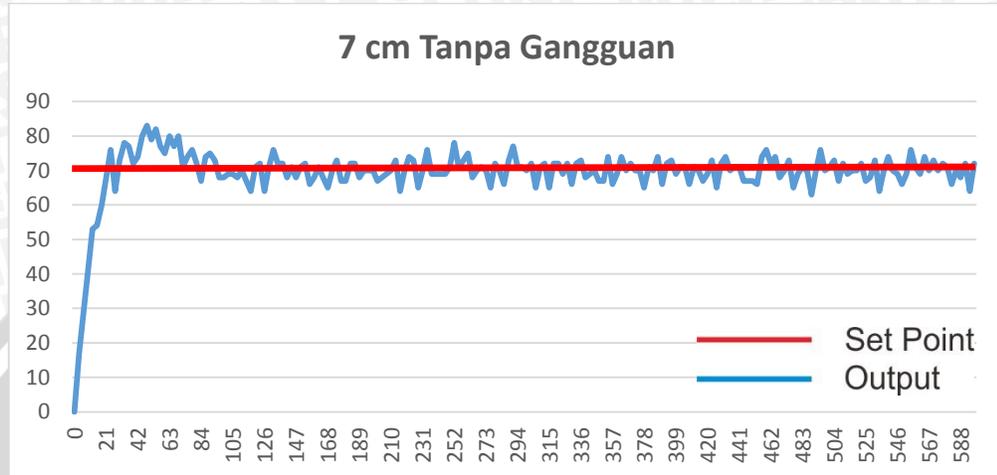
- Rangkaian sistem keseluruhan
- Perangkat komputer
- Power supply 12 V

c. Langkah Pengujian

1. Merangkai semua perangkat pada *plant* tangki air (73426)
2. Memasukkan *set point*: 7 – 9 cm dengan kenaikan 1cm melalui program arduino IDE
3. Percobaan menggunakan gangguan dengan memasukkan sekat-sekat berlubang pada tangki pengisian saat sistem sudah *steady*
4. Mencatat hasil pembacaan ketinggian air melalui serial monitor program arduino dan ditampilkan melalui grafik

d. Hasil Pengujian

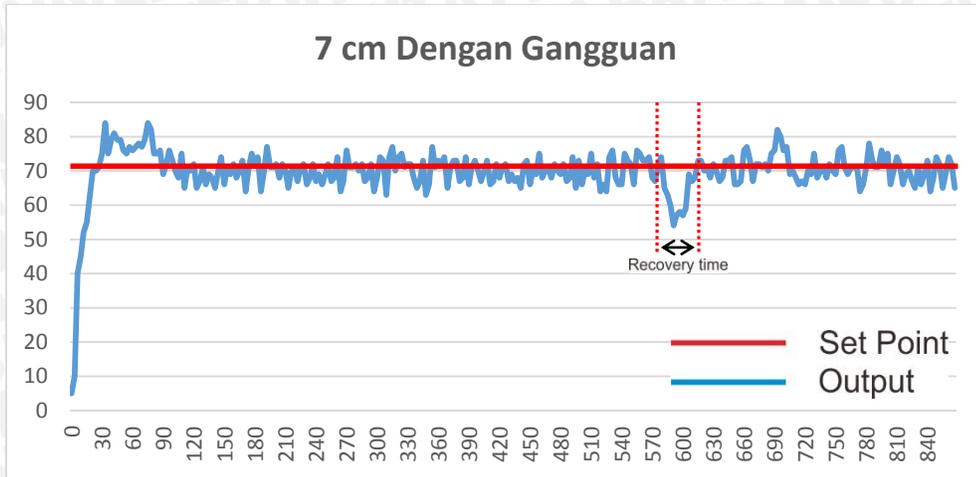
Hasil pengujian *plant* tangki air 73426 dengan *set point* 7 cm dan periode sampling 3 detik ditunjukkan dalam Gambar 5.2.



Gambar 5.2 Grafik *Output Plant* dengan *Set Point* 7cm Tanpa Gangguan

Dari grafik *output plant* dalam Gambar 5.2 dapat disimpulkan bahwa parameter PID yang diimplementasikan pada alat dengan *setpoint* ketinggian 7 cm memiliki *error steady state* sebesar 2,28%, maksimal *overshoot* (M_p) 17,1%, *settling time* sebesar 93 detik, *rise time* (T_r) 21 detik, *delay time* (T_d) 9 detik dan *peak time* (T_p) 48 detik.

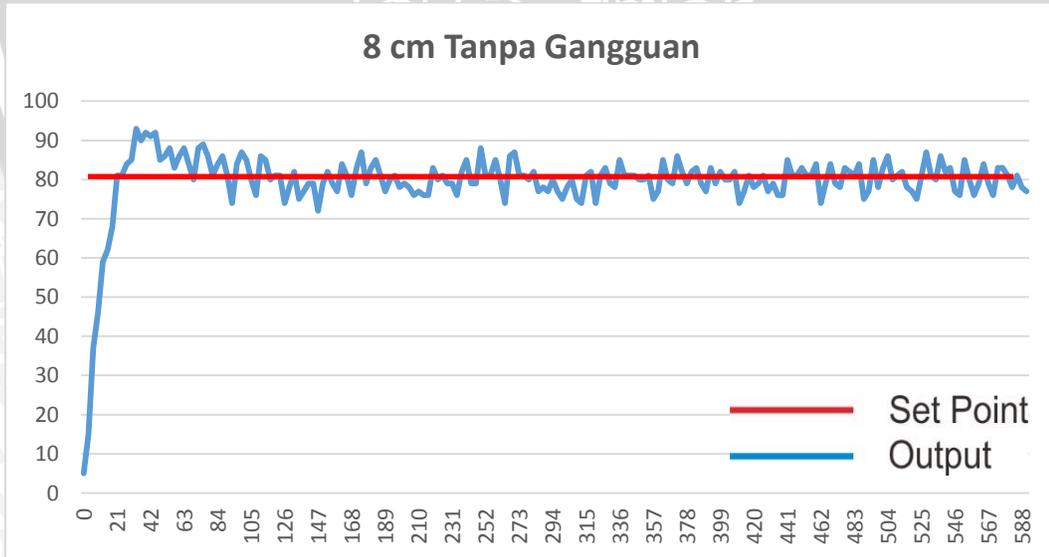
Hasil pengujian *Plant* tangki air 73426 pada *set point* 7 cm dan periode sampling 3 detik dengan gangguan berupa pemberian sekat berlubang pada detik ke 585 ditunjukkan dalam Gambar 5.3.



Gambar 5.3 Grafik Output Plant Set Point 7cm dengan Gangguan

Dari grafik *output plant* dalam Gambar 5.3 dapat disimpulkan bahwa parameter PID yang diimplementasikan pada alat dengan *setpoint* ketinggian 7 cm diberi gangguan, memiliki *error steady state* sebesar 3,1%, maksimal *overshoot* (M_p) 18,5%, *settling time* sebesar 90 detik, *recovery time* 21 detik, *rise time* (T_r) 24 detik, *delay time* (T_d) 6 detik dan *peak time* (T_p) 42 detik.

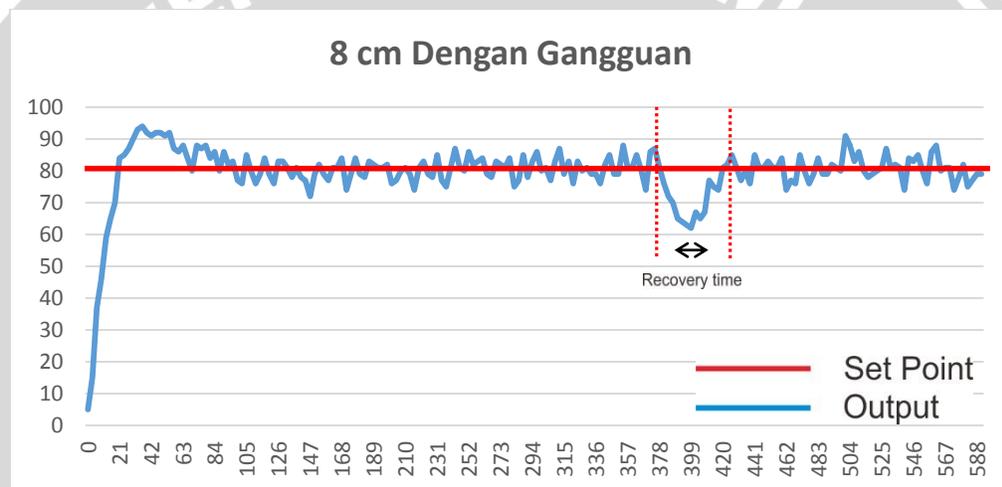
Hasil pengujian *plant* tangki air 73426 dengan set point 8 cm dan periode sampling 3 detik ditunjukkan dalam Gambar 5.4.



Gambar 5.4 Grafik Output Plant dengan Set Point 8 cm Tanpa Gangguan

Dari grafik *output plant* dalam Gambar 5.4 dapat disimpulkan bahwa parameter PID yang diimplementasikan pada alat dengan *setpoint* ketinggian 8 cm memiliki *error steady state* sebesar 2.5%, maksimal *overshoot* (M_p) 15%, *settling time* sebesar 105 detik, *rise time* (T_r) 24 detik, *delay time* (T_d) 9 detik dan *peak time* (T_p) 39 detik.

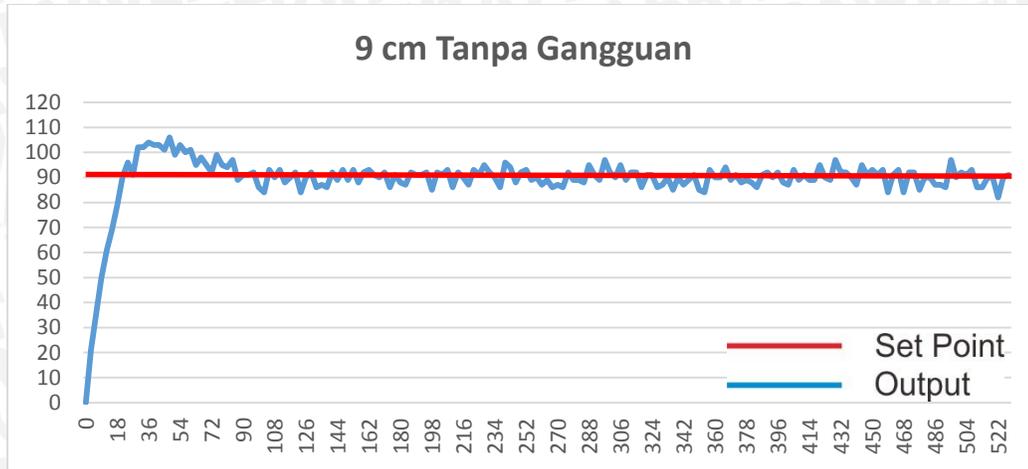
Hasil pengujian *Plant* tangki air 73426 pada *set point* 8 cm dan periode *sampling* 3 detik dengan gangguan berupa pemberian sekat berlubang pada detik ke 384 ditunjukkan dalam Gambar 5.5.



Gambar 5.5 Grafik *Output Plant* Set Point 8 cm dengan Gangguan

Dari grafik *output plant* dalam Gambar 5.5 dapat disimpulkan bahwa parameter PID yang diimplementasikan pada alat dengan *set point* ketinggian 8 cm diberi gangguan, memiliki *error steady state* sebesar 3.4%, maksimal *overshoot* (M_p) 16,2%, *settling time* sebesar 108 detik, *recovery time* 24 detik, *rise time* (T_r) 21 detik, *delay time* (T_d) 6 detik dan *peak time* (T_p) 39 detik.

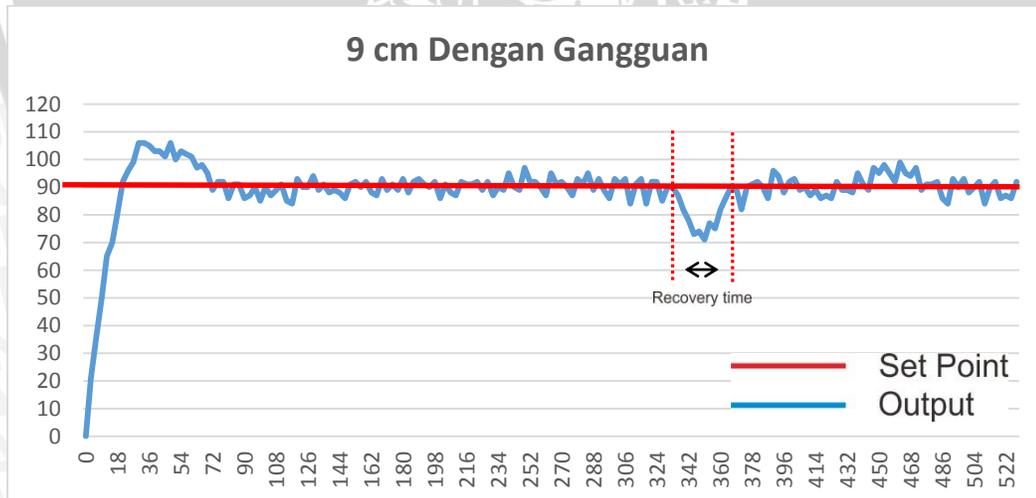
Hasil pengujian *plant* tangki air 73426 dengan *set point* 9 cm dan periode *sampling* 3 detik dalam Gambar 5.6.



Gambar 5.6 Grafik Output Plant dengan Set Point 9 cm tanpa Gangguan

Dari grafik *output plant* dalam Gambar 5.6 dapat disimpulkan bahwa parameter PID yang diimplementasikan pada alat dengan *set point* ketinggian 9 cm, memiliki *error steady state* sebesar 2.3%, maksimal *overshoot* (M_p) 16,6%, *settling time* sebesar 108 detik, *rise time* (T_r) 21 detik, *delay time* (T_d) 9 detik dan *peak time* (T_p) 48 detik.

Hasil pengujian *Plant* tangki air 73426 pada *set point* 9 cm dan periode sampling 3 detik dengan gangguan berupa pemberian sekat berlubang pada detik ke 339 ditunjukkan dalam Gambar 5.7.



Gambar 5.7 Grafik Output Plant Set Point 9 cm dengan Gangguan

Grafik *output plant* dalam Gambar 5.7 menunjukkan bahwa parameter PID yang diimplementasikan pada alat dengan *set point* ketinggian 9 cm diberi gangguan, memiliki *error steady state* sebesar 3,07%, maksimal *overshoot* (M_p) 18,6%, *settling time* sebesar 108 detik, *recovery time* 24 detik, *rise time* (T_r) 21 detik, *delay time* (T_d) 9 detik dan *peak time* (T_p) 39 detik.



UNIVERSITAS BRAWIJAYA



BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

1. Perancangan kontroler terdiri atas perancangan perangkat keras yaitu merancang modul mikrokontroler beserta driver dan modul sensor. Kemudian melakukan perancangan perangkat lunak yang berupa pembuatan program pada *software* mikrokontroler untuk pembacaan sensor dan aksi pompa.
2. Pencarian parameter menggunakan metode pertama Ziegler – Nichols dengan cara pemberian masukan berupa unit step dan mendapatkan kurva respon berbentuk huruf S, kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda (L) dan konstanta waktu (T), dilanjutkan dengan pencarian parameter kontrol PID melalui tabel penalaan Ziegler – Nichols sehingga didapatkan parameter kontrol PID sebesar $K_p = 3,46$ $K_i = 0,96$ dan $K_d = 3,11$.
3. Hasil perancangan kontroler PID pada *plant* tangki air 73426 diterapkan pada 3 *set point*.

Tabel 6.1. Hasil Pengujian Keseluruhan Sistem

<i>Set Point</i>	<i>Settling Time</i>	<i>Error Steady State</i>	<i>Overshoot</i>	<i>Recovery Time</i>
7 cm	93 detik	2,28%	17,1%	-
7 cm dengan Gangguan	90 detik	3,1%	18,5%	21 detik
8 cm	105 detik	2,5%	15%	-
8 cm dengan Gangguan	108 detik	3,4%	16,2%	24 detik
9 cm	108 detik	2,3%	16,6%	-
9 cm dengan Gangguan	108 detik	3,07%	18,6%	24 detik

6.2 Saran

Mengaplikasikan pengontrolan *plant* tangki air 73426 ke *plant* pengontrolan ketinggian air untuk kehidupan sehari-hari maupun untuk bidang industri.

