



**TUGAS AKHIR TF 145565**

**RANCANG BANGUN SISTEM PENGENDALIAN  
LEVEL PADA TANGKI PEMANAS *PLANT*  
*FLUIDIZED BED REACTOR* PENGOLAHAN *OIL*  
*SLUDGE***

**Putri Merlisa Wahyuningrum  
NRP. 2414 031 027**

**Dosen Pembimbing  
Totok Ruki Biyanto, ST, MT, Ph.D  
NIP. 19710702 199802 1 001**

**DEPARTEMEN TEKNIK INSTRUMENTASI  
FAKULTAS VOKASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2017**



**TUGAS AKHIR – TF 145565**

**RANCANG BANGUN SISTEM  
PENGENDALIAN *LEVEL* PADA TANGKI  
PEMANAS *PLANT FLUIDIZED BED*  
*REACTOR* PENGOLAHAN *OIL SLUDGE***

**PUTRI MERLISA WAHYUNINGRUM  
NRP. 2414 031 027**

Dosen Pembimbing  
Totok Ruki Biyanto, ST. MT. Ph.D  
NIP. 19710702 199802 1 001

**DEPARTEMEN TEKNIK INSTRUMENTASI  
FAKULTAS VOKASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2017**



*FINAL PROJECT – TF 145565*

***DESIGN OF LEVEL CONTROL SYSTEM AT  
HEATING TANK PLANT FLUIDIZED BED  
REACTOR OIL SLUDGE PROCESS***

*PUTRI MERLISA WAHYUNINGRUM  
NRP. 2414 031 027*

*Supervisor  
Totok Ruki Biyanto, ST. MT. Ph.D  
NIP. 19710702 199802 1 001*

*DEPARTMENT OF INSTRUMENTATION ENGINEERING  
FACULTY OF VOCATION  
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY  
SURABAYA 2017*

**RANCANG BANGUN SISTEM PENGENDALIAN  
LEVEL PADA TANGKI PEMANAS PLANT  
FLUIDIZED BED REACTOR PENGOLAHAN OIL  
SLUDGE**

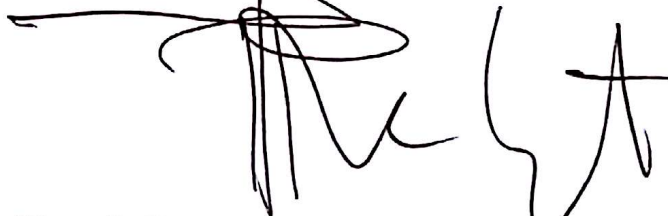
**TUGAS AKHIR**

**Oleh :**

**Putri Merlisa Wahyuningrum  
NRP. 2414 031 027**

**Surabaya, 26 Juli 2017  
Mengetahui / Menyetujui**

**Dosen Pembimbing**



**Totok Ruki Biyanto, S.T, M.T, Ph.D  
NIP. 19710702 199802 1 001**

**Kepala Departemen  
Teknik Instrumentasi FV – ITS**



**Dr. Ir. Purwadi Agus D, M.Sc.  
NIP. 19620822 198803 1 001**

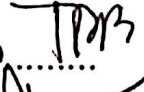



**RANCANG BANGUN SISTEM PENGENDALIAN LEVEL  
PADA TANGKI PEMANAS PLANT FLUIDIZED BED  
REACTOR PENGOLAHAN OIL SLUDGE**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Ahli Madya  
pada  
Program Studi D3 Teknik Instrumentasi  
Departemen Teknik Instrumentasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

Oleh :  
**PUTRI MERLISA WAHYUNINGRUM**  
NRP. 2414 031 027

Disetujui oleh Tim Penguji Tugas Akhir :

1. Totok Ruki Biyanto, S.T, M.T, Ph.D..... (Pembimbing)
2. Gunawan Nugroho, S.T, M.T, Ph.D ..... (Ketua Tim Penguji)
3. Hendra Cordova, S.T, M.T ..... (Penguji II)
4. Murry Raditya, S.T, M.T ..... (Penguji III)

**SURABAYA**  
**Juli 2017**

# **RANCANG BANGUN SISTEM PENGENDALIAN LEVEL PADA TANGKI PEMANAS PLANT FLUIDIZED BED REACTOR**

*Nama* : Putri Merlisa Wahyuningrum  
*NRP* : 2414031027  
*Departemen* : Teknik Instrumentasi, FV - ITS  
*Dosen Pembimbing* : Totok Ruki Biyanto, ST. MT. Ph.D

## **Abstrak**

*Kebutuhan energi yang besar, menyebabkan banyak perusahaan minyak dan gas meningkatkan aktifitas eksplorasi dan produksinya. Saat minyak diekplorasi dan diproduksi dari sumur, maka sludge akan ikut terbawa. Sludge merupakan limbah dari minyak mentah yang masih banyak mengandung pasir, tanah, dan serat-serat atau ampas yang terikut di dalam minyak. Salah satu proses pengambilan minyak pada sludge adalah memberikan steam pada sludge sehingga minyak bisa terlepas dari sludge dan larut di dalam uap air. Untuk proses tersebut diperlukan pengendalian level untuk menjaga ketinggian fluida pada tangki pemanas agar didapatkan steam yang sesuai. Sistem pengendalian ini menggunakan sensor level switch dan mikrokontroler ATmega 32 sebagai controller. Aktuator yang digunakan berupa motor servo. Hasil respon sistem proses tersebut pada open loop, dapat dilihat bahwa proses tersebut memerlukan controller untuk menjamin agar proses dapat berjalan dengan baik. Pada closed loop, dapat diketahui bahwa sistem dari tangki pemanas membutuhkan pengendalian agar dapat menjaga set point sebesar 10 cm dengan rise time sebesar 22,52 sekon dan pada uji disturbance diketahui bahwa performansi pengendali tersebut dalam mempertahankan set point terhadap perubahan dengan baik.*

**Kata Kunci** : Sistem Pengendalian Level, Open Loop, Closed Loop, Disturbance

# **DESIGN OF LEVEL CONTROL SYSTEM AT HEATING TANK OIL SLUDGE PROCESS PLANT FLUIDIZED BED REACTOR**

**Name** : Putri Merlisa Wahyuningrum  
**NRP** : 2414031027  
**Department** : Instrumentation Engineering, FV - ITS  
**Supervisor** : Totok Ruki Biyanto, ST. MT. Ph.D

## **Abstract**

*Large energy needs, causing many oil and gas companies to increase their exploration and production activities. When oil explored and produced from well, sludge will be carried along. Sludge is a waste of crude oil that still contains a lot of sand, soil, and fibers or pulp in oil. One of the process of taking oil in sludge is to provide steam to sludge so that oil can be released from sludge and dissolve in water vapor. For this process level control is required to maintain the height of the fluid in the heating tank to obtain the appropriate steam. This control system uses Level Switch sensors and ATmega 32 microcontroller as a controller. The actuator used is a servo motor. The results of the process system response to the open loop, can be seen that the process requires a controller to ensure that the process can run well. In the closed loop, it can be seen that the system of the heating tank requires control in order to maintain the set point value is 10 cm with the rise time of 22.52 seconds and on the disturbance test it is known that the performance of the controller in maintaining the set point to change well.*

**Keywords** : Level Control, Open Loop, Closed Loop, Disturbance

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT atas limpahan rahmat dan hidayah-Nya serta shalawat dan salam kepada Nabi Muhammad SAW sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“Rancang Bangun Sistem Pengendalian Level pada Tangki Pemanas Plant Fluidized Bed Reactor Pengolahan Oil Sludge”**. Penulis telah banyak mendapatkan bantuan dari berbagai pihak dalam menyelesaikan Tugas Akhir ini. Untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Dr. Ir. Purwadi Agus Darwito, M.Sc selaku Ketua Departemen Teknik Instrumentasi ITS dan Bapak Wiratno Argo Asmoro, M.Sc selaku dosen wali penulis yang telah sabar memberikan dukungan, bimbingan, serta ilmu yang sangat bermanfaat.
2. Bapak Totok Ruki Biyanto, ST, MT, Ph.D selaku dosen pembimbing yang senantiasa memberikan motivasi, bimbingan dan arahan dalam menyelesaikan Tugas Akhir ini.
3. Bapak dan Ibu dosen Teknik Instrumentasi yang telah memberikan ilmu selama kuliah.
4. Seluruh Staf Jurusan Teknik Instrumentasi yang telah membantu penulis dalam hal administrasi.
5. Ibu tercinta Ambar Kusumaningsih, Bapak Yosep Cahyono dan seluruh keluarga besar tercinta yang senantiasa memberikan dukungan, semangat dan do'a kepada penulis.
6. Teman-teman seperjuangan dalam mengerjakan Tugas Akhir Tim *Fluidized Bed Reactor* (Aloys Yantri Martina dan Agung Marhamsyah).
7. Sahabat Zelena yang senantiasa menemani, Tim Garam (Imam, Syam, Ninis dan Alam), Tim Biogas (Wahyu, Devi, Indro, Okke dan Layly), Tim Splitting dan Mixing Piping (Nova dan Fery), Tim Heating Flask ( Nada dan Heni), Tim Detergen (Dede dan Bagas), Tim PV (Aulia,



Mufit, Iren, Izzah, Yunita dan Dimas), Tim Kunyit (Ifa dan Ubai), Tim Monitoring Gas (Kacong, Mufida, Nawang dan Haryo) serta Tim Robot (Amanda, Haqqi, Bisma dan Roni)

8. Workshop Instrumentasi yang telah meminjamkan alat-alatnya dan orang-orang yang setia membantu pada saat hari H (Maul, Tantowi dan Amin)
9. Teman-teman D3 Teknik Instrumentasi 2014 lainnya serta teman-teman S1 Teknik Fisika angkatan 2014 yang selalu memotivasi penulis.

Penulis menyadari bahwa penulisan laporan Tugas Akhir ini tidaklah sempurna. Oleh karena itu sangat diharapkan kritik dan saran yang membangun dari semua pihak sehingga mencapai sesuatu yang lebih baik lagi. Penulis juga berharap semoga laporan ini dapat menambah wawasan yang bermanfaat bagi pembacanya.

Surabaya, 26 Juli 2017

Penulis.

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PENGESAHAN</b> .....	iii
<b>ABSTRAK</b> .....	v
<b>ABSTRACT</b> .....	vi
<b>KATA PENGANTAR</b> .....	vii
<b>DAFTAR ISI</b> .....	ix
<b>DAFTAR GAMBAR</b> .....	xi
<b>DAFTAR TABEL</b> .....	xii

### **BAB I. PENDAHULUAN**

1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Lingkup Kajian .....	2
1.4 Tujuan .....	3
1.5 Manfaat .....	3

### **BAB II. TINJAUAN PUSTAKA**

2.1 <i>Oil Sludge</i> .....	5
2.2 Sistem Pengendalian .....	6
2.3 <i>Sensor Level Switch</i> .....	7
2.4 Motor Servo .....	8
2.5 Ketel Uap.....	9
2.6 <i>Fluidized Bed Reactor</i> .....	10
2.7 <i>Storage Tank</i> .....	11
2.8 <i>Ball Valve</i> .....	12
2.9 Mikrokontroler Atmega32 .....	13
2.10 <i>Liquid Crystal Display (LCD)</i> .....	15
2.11 <i>Heater</i> .....	16

### **BAB III. PERANCANGAN DAN PEMBUATAN ALAT**

3.1 Diagram Alir Perancangan dan Pembuatan Alat .....	19
3.2 Studi Literatur .....	20
3.3 Perancangan Sistem dan Pembuatan Alat .....	20
3.3.1. Pembuatan Mekanik .....	20

3.3.2. Perancangan <i>Hardware</i> dan <i>Software</i> .....	25
3.4 Integrasi.....	29
3.5 Pengujian Alat.....	29
3.6 Pengambilan Data .....	30
3.7 Analisa dan Pembahasan.....	31
3.8 Penulisan laporan .....	31

**BAB IV. PENGUJIAN ALAT DAN PEMBAHASAN**

4.1 Pengujian Sensor <i>Level Switch</i> .....	33
4.2 Respon Sistem.....	37
4.2.1 <i>Open Loop</i> .....	37
4.2.2 <i>Close Loop</i> .....	38
4.3 Uji <i>Disturbance</i> .....	40
4.4 Pembahasan.....	42

**BAB V. PENUTUP**

5.1 Kesimpulan .....	45
5.2 Saran .....	45

**DAFTAR PUSTAKA**

LAMPIRAN A ( <i>Datasheet Sensor Level Switch</i> ) .....	A-1
LAMPIRAN B ( <i>Datasheet Mikrokontroler Atmega 32</i> ) ...	B-1
LAMPIRAN C ( <i>Datasheet Motor Servo MG996R Tower Pro</i> ).....	C-1
LAMPIRAN D ( <i>Listing Program</i> ) .....	D-1

**BIOGRAFI PENULIS**

## DAFTAR GAMBAR

<b>Gambar 2.1</b>	<i>Oil Sludge</i> .....	5
<b>Gambar 2.2</b>	Diagram Blok Sistem Pengendalian Otomatis .....	7
<b>Gambar 2.3</b>	Sensor <i>Level Switch</i> .....	7
<b>Gambar 2.4</b>	Motor Servo MG 996R Tower Pro .....	9
<b>Gambar 2.5</b>	Ketel Uap .....	10
<b>Gambar 2.6</b>	<i>Fluidized Bed Reactor</i> .....	11
<b>Gambar 2.7</b>	<i>Storage Tank</i> .....	12
<b>Gambar 2.8</b>	<i>Ball Valve</i> .....	13
<b>Gambar 2.9</b>	Konfigurasi Pin Mikrokontroler Atmega 32 .....	14
<b>Gambar 2.10</b>	<i>Liquid Crystal Display</i> .....	16
<b>Gambar 3.1</b>	Diagram Alir Tugas Akhir .....	17
<b>Gambar 3.2</b>	P&ID <i>Plant Fluidized Bed Reactor</i> .....	20
<b>Gambar 3.3</b>	P&ID Sistem Pengendalian Level.....	20
<b>Gambar 3.4</b>	<i>Plant Fluidized Bed Reactor</i> .....	21
<b>Gambar 3.5</b>	Tangki Pemanas .....	22
<b>Gambar 3.6</b>	Diagram Blok Pengendalian Level .....	22
<b>Gambar 3.7</b>	Konfigurasi Sensor <i>Level Switch</i> .....	24
<b>Gambar 3.8</b>	Konfigurasi Motor Servo .....	25
<b>Gambar 3.9</b>	Konfigurasi <i>Display LCD</i> .....	25
<b>Gambar 3.10</b>	Listing Program pada CVAVR .....	26
<b>Gambar 3.11</b>	Tampilan pada Visual Studio .....	27
<b>Gambar 4.1</b>	<i>Plant Fluidized Bed Reactor</i> .....	34
<b>Gambar 4.2</b>	Grafik Tegangan Saat Kondisi <i>High</i> .....	35
<b>Gambar 4.3</b>	Grafik Tegangan Saat Kondisi <i>Low</i> .....	36
<b>Gambar 4.4</b>	Sistem Pengendalian Terbuka Pada Tangki .....	37
<b>Gambar 4.5</b>	Grafik <i>Open Loop</i> Sistem Pengendalian <i>Level</i> ...	38
<b>Gambar 4.6</b>	Diagram Blok Sistem Pengendalian <i>Level</i> .....	39
<b>Gambar 4.7</b>	Grafik <i>Close Loop</i> Sistem Pengendalian <i>Level</i> ...	40
<b>Gambar 4.8</b>	<i>Uji Disturbance</i> .....	42

## DAFTAR TABEL

<b>Tabel 3.1</b> Tabel Spesifikasi Rancang Bangun .....	19
<b>Tabel 4.1</b> Hasil pengujian Sensor <i>Level Switch High</i> .....	34
<b>Tabel 4.2</b> Hasil pengujian Sensor <i>Level Switch Low</i> .....	36
<b>Tabel 4.3</b> Hasil Pengujian <i>Open Loop</i> .....	37
<b>Tabel 4.4</b> Hasil Pengujian <i>Close Loop</i> .....	39
<b>Tabel 4.5</b> Hasil Pengujian <i>Disturbance</i> .....	41

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Banyak perusahaan minyak dan gas, perusahaan tambang ataupun perusahaan industri lainnya yang melakukan eksplorasi dan eksploitasi sumber daya alam baik di lautan maupun di daratan untuk diolah menjadi produk yang dapat memenuhi kebutuhan masyarakat. Pada proses pengolahan sumber daya alam tersebut akan menghasilkan produk samping berupa limbah. Limbah merupakan buangan dalam bentuk zat cair atau padat yang mengandung bahan berbahaya dan beracun karena sifat dan konsentrasinya atau jumlahnya, baik secara langsung maupun tidak langsung dapat mencemari atau merusak lingkungan hidup dan membahayakan kesehatan, kelangsungan hidup manusia maupun makhluk hidup lainnya.

Kebutuhan energi yang besar, khususnya minyak dan gas menyebabkan banyak perusahaan minyak dan gas meningkatkan aktifitas eksplorasi dan produksinya. Saat minyak dieksplorasi dan diproduksi dari sumur, maka *sludge* akan ikut terbawa. *Sludge* merupakan limbah dari minyak mentah yang berasal dari *Crude Oil Tank* (COT) yang masih banyak mengandung pasir, tanah, dan serat-serat atau ampas yang terikut di dalam minyak. Setiap tahun, perusahaan minyak dan gas seperti Pertamina harus membuang ratusan bahkan ribuan ton *sludge* dengan biaya pembuangan 75 US tiap ton.<sup>[1]</sup> Karena alasan biaya yang mahal, limbah ini hanya ditimbun pada gudang-gudang penyimpan limbah tanpa adanya proses pengolahan yang memadai. Padahal kandungan *sludge* pada minyak masih dapat diolah menjadi produk yang berguna. Namun disamping itu, adanya *sludge* dapat mengakibatkan rusaknya minyak pada saat penyimpanan sehingga menyebabkan menurunnya mutu minyak sehingga diperlukan pemisahan *sludge* dari minyak. Pengolahan *sludge* biasanya dengan mengambil kandungan minyak dalam *sludge* sampai batas minimum agar *sludge* dianggap ramah lingkungan. Untuk itu diperlukan proses yang sesuai untuk mengambil minyak dari *sludge*. Salah satu

proses pengambilan minyak pada *sludge* adalah memberikan steam pada *sludge* sehingga minyak bisa terlepas dari *sludge* dan larut di dalam uap air. Untuk proses tersebut diperlukan *level* air yang sesuai agar pemisahan minyak dari *sludge* bisa terjadi. *Level* air ini bisa dipenuhi dengan memasang pengendali *level* yang terdiri dari pengukuran *level* air, *controller* dan *control valve*. Pada Tugas Akhir ini, dirancang pengendalian *level* air yang akan menjadi uap untuk memenuhi keperluan proses pemisahan minyak dari *sludge* dengan membuat sebuah tangki pemanas sistem pengendalian *level* air pada *plant Fluidized Bed Reactor* pengolahan *oil sludge*.

### 1.2 Rumusan Masalah

Berdasarkan latar belakang yang dijelaskan diatas, maka rumusan masalah dalam Tugas Akhir ini adalah :

1. Perlu dirancang sistem pengendalian *level* air pada tangki pemanas *plant Fluidized Bed Reactor*.
2. Bagaimana hasil performansi sistem pengendalian *level* air pada tangki pemanas *plant Fluidized Bed Reactor*?

### 1.3 Lingkup Kajian

Kajian ini dimulai dari penentuan *level* air yang diperlukan pada proses pemisahan minyak dari *sludge*. Proses pembentukan *steam* dimulai dari penyimpanan air yang dipompa ke dalam pemanas sehingga berbentuk uap. Perancangan dan pembuatan instrumentasi pengukuran *level* air dimulai dari sensor *Level Switch*, *controller* berbasis *Mikrokontroller Atmega32* dan *actuator* berupa motor servo.

### 1.4 Tujuan

Tujuan dalam pelaksanaan Tugas Akhir ini antara lain :

1. Untuk merancang alat sistem pengendalian *level* air pada tangki pemanas *plant Fluidized Bed Reactor*.
2. Untuk mengetahui hasil performansi sistem pengendalian *level* air pada tangki pemanas *plant Fluidized Bed Reactor*.

### **1.5 Manfaat**

Manfaat dari Tugas Akhir ini adalah sebagai sistem pengendalian *level* air pada *plant Fluidized Bed Reactor* pengolahan *oil sludge* pada tangki pemanas menggunakan *mikrokontroller* Atmega 32 yang mana diharapkan Tugas Akhir ini nantinya dapat dijadikan media pembelajaran dan pengetahuan bagi mahasiswa.



*Halaman ini sengaja dikosongkan*

## BAB II TINJAUAN PUSTAKA

### 2.1 *Oil Sludge*

*Oil sludge* adalah endapan berupa lumpur atau pasta berwarna hitam yang tercampur dengan tanah, kerikil, air dan bahan lainnya. Endapan tersebut semakin lama semakin menumpuk pada bagian bawah dari tangki-tangki penyimpanan atau pada pipa-pipa penyaluran bahan bakar minyak sehingga dapat menyumbat pipa, mengurangi kapasitas operasional dari tangki penyimpanan minyak serta mempercepat proses korosi (Banat and Rancich, 2009).



**Gambar 2.1** *Oil Sludge*

Biasanya *oil sludge* masih mengandung minyak antara 20-40%. Temperatur yang cukup ( $460^{\circ}\text{C}$  -  $650^{\circ}\text{C}$ ) akan memudahkan proses pemisahan minyak dari *sludge*. Untuk memisahkan minyak dari *sludge* diperlukan sejumlah energi dan *level* air tertentu. Untuk mencapai titik didih tersebut diperlukan pemanas agar mencapai temperatur yang sesuai. Jumlah energi yang diperlukan untuk sejumlah *sludge* tergantung banyaknya *sludge* dan kandungan minyaknya. Untuk memenuhi kesetimbangan untuk menguapkan minyak dari *sludge* diperlukan sejumlah massa *steam* sebesar :

$$Q = m.c.\Delta t \quad (2.1)$$

Dimana :

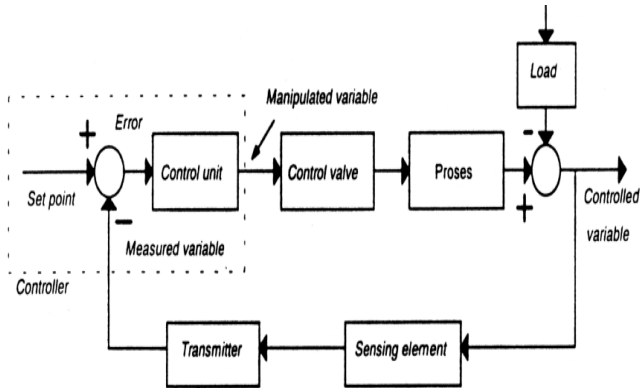
- Q = Kalor (Joule)  
 m = massa air (kg)  
 c = kalor jenis (J/kg K)  
 $\Delta t$  = perubahan suhu (K)

## 2.2 Sistem Pengendalian

Instrumentasi adalah ilmu yang mempelajari tentang penggunaan peralatan atau instrument untuk mengukur dan mengatur suatu besaran baik kondisi fisis maupun kimia. Dari definisi tersebut dapat diambil suatu kesimpulan atau prinsip dasar, bahwa instrumentasi terdiri dari dua pokok kegiatan yaitu mengukur dan mengatur suatu besaran. Operasi di industri seperti kilang minyak (*refinery*) dan petrokimia (*petrochemical*) sangat bergantung pada pengukuran dan pengendalian besaran proses. Beberapa besaran proses yang harus diukur dan dikendalikan pada suatu industri proses, misalnya aliran (*flow*) di dalam pipa, tekanan (*pressure*) di dalam sebuah *vessel*, suhu (*temperature*) di *heat exchanger*, serta permukaan (*level*) zat cair di sebuah tangki.

Sistem pengendalian bertujuan untuk mengontrol proses atau *plant* yang terjadi di industri. Operasi yang berjalan di industri ini dapat bersifat diskret (*on-off*), misal *valve* terbuka atau tertutup, motor hidup atau mati, konveyor jalan atau berhenti, dan lain-lain. Selain itu juga secara kontinyu, suatu pengaturan untuk mempertahankan tinggi cairan dalam tanki pada nilai tertentu.

Di dalam diagram blok sistem pengendalian otomatis, terdapat komponen-komponen pokok seperti elemen proses, elemen pengukuran (*sensing element* dan *transmitter*), elemen *controller* (*control unit*) dan *final control element*. Di dalam gambar bagian atas *controller* mempunyai *summing junction* dengan tanda positif dan negatif. Di titik inilah langkah membandingkan dilakukan dengan mengurangi besaran *set point* dengan sinyal *measurement variable*. Hasilnya adalah sinyal *error*.



**Gambar 2.2** Diagram Blok Sistem Pengendalian Otomatis

### 2.3 Sensor Level Switch

*Level Switch* merupakan saklar otomatis yang digunakan untuk mendeteksi ketinggian contohnya digunakan untuk mendeteksi suatu volume benda cair yang terdapat pada suatu tabung atau tangki penampungan seperti tangki air, tangki minyak dll. Prinsip kerjanya seperti pada sistem pengendali otomatis secara *on-off* dimana terdapat batas atas dan batas bawah dengan range yang ditentukan. Batas atas dan batas bawah ini ditentukan oleh pelampung yang terbuat dari plastik yang menempel pada batang besi yang ketinggiannya dapat diatur sesuai keinginan. Apabila ketinggian air di bawah *level switch* ini maka pelampung berada pada batas bawahnya dan ketika ketinggian cairan meningkat maka akan membuat pelampung ini naik hingga batas atasnya.

Sensor dari *level switch* berupa besi panjang sementara rangkaian kontrolnya berbentuk bulat yang terdapat di top feed bin. Rangkaian kontrol bagian dalam terdapat rangkaian elektronik berupa terminal yang dihubungkan ke perangkat listrik, ada 8 terminal dan 7 kabel. Namun yang dipasang hanya 4 kabel, ke-4 ini terdiri dari: kabel power positif (+) dan negatif (-), kabel pendeteksi *low*, kabel pendeteksi *high*. *Level Switch* mempunyai

tegangan kerja antara 100-200 VAC dan mempunyai beban kerja sekitar 5 Ampere.



**Gambar 2.3** *Level Switch*

## 2.4 Motor Servo

Motor servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik *loop* tertutup (servo), sehingga dapat di set-up atau di atur untuk menentukan dan memastikan posisi sudut dari poros output motor untuk menggerakkan *valve*. Motor servo merupakan perangkat yang terdiri dari motor DC, serangkaian *gear*, rangkaian kontrol dan potensiometer. Serangkaian *gear* yang melekat pada poros motor DC akan memperlambat putaran poros dan meningkatkan torsi motor servo, sedangkan potensiometer dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo.

Ada dua jenis motor servo, yaitu motor servo AC dan DC. Motor servo AC lebih dapat menangani arus yang tinggi atau beban berat, sehingga sering diaplikasikan pada mesin-mesin industri. Sedangkan motor servo DC digunakan pada aplikasi-aplikasi yang lebih kecil. Dan bila dibedakan menurut rotasinya, umumnya terdapat dua jenis motor servo yang terdapat di pasaran, yaitu *motor servo rotation 180°* dan *servo rotation continuous 360°*.

- a. Motor servo *standard (servo rotation 180°)* adalah jenis yang paling umum dari motor servo, dimana putaran poros

outputnya terbatas hanya  $90^\circ$  kearah kanan dan  $90^\circ$  kearah kiri. Dengan kata lain total putarannya hanya setengah lingkaran atau  $180^\circ$ .

b. Motor servo *rotation continuous*  $360^\circ$  merupakan jenis motor servo yang sama dengan jenis servo *standard*, perputaran porosnya tanpa batasan dengan kata lain dapat berputar terus, baik ke arah kanan maupun kiri.<sup>[3]</sup>



**Gambar 2.4** Motor Servo MG 996R Tower Pro

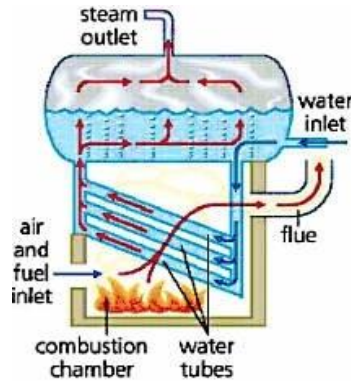
## 2.5 Ketel Uap

Boiler atau ketel uap adalah suatu bejana/wadah yang di dalamnya berisi air atau fluida lain untuk dipanaskan. Energi panas dari fluida tersebut selanjutnya digunakan untuk berbagai macam keperluan, seperti untuk turbin uap, pemanas ruangan, mesin uap, dan lain sebagainya. Secara proses konversi energi, boiler memiliki fungsi untuk mengkonversi energi kimia yang tersimpan di dalam bahan bakar menjadi energi panas yang tertransfer ke fluida kerja.

Bejana bertekanan pada boiler umumnya menggunakan bahan baja dengan spesifikasi tertentu yang telah ditentukan dalam standard ASME (*The ASME Code Boilers*), terutama untuk penggunaan boiler pada industri-industri besar. Terdapat berbagai macam jenis material yang digunakan sebagai bahan pembuatan boiler seperti tembaga, kuningan, dan besi cor.

Panas yang diberikan kepada fluida di dalam boiler berasal dari proses pembakaran dengan berbagai macam jenis bahan bakar yang dapat digunakan, seperti kayu, batubara, solar/minyak bumi, dan gas dan jumlah produksi uap tergantung pada luas permukaan

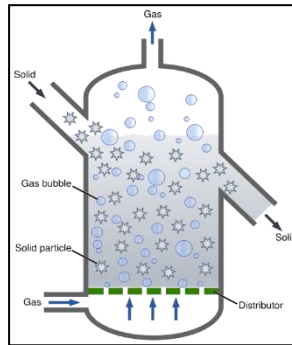
pemindah panas, laju aliran, dan panas pembakaran yang diberikan.<sup>[4]</sup>



**Gambar 2.5** Ketel Uap

## 2.6 *Fluidized Bed Reactor*

Fluidisasi merupakan salah satu bentuk peristiwa dimana partikel berfase padatan diubah menjadi fase yang memiliki perilaku layaknya fluida cair dengan cara diberi kontak dengan gas atau cairan. Fenomena fluidisasi ini berlangsung pada media yang dikenal dengan sebutan *fluidized bed reactor*. *Fluidized bed reactor* merupakan suatu media yang berbentuk bejana yang berisikan partikel berfase padat yang kemudian akan dialiri oleh fluida hingga terfluidisasi. Peristiwa fluidisasi ini terjadi akibat adanya gaya drag yang bekerja pada partikel berfase padat yang mana disebabkan karena aliran fluida yang bergerak dari bawah ke atas yang lebih besar daripada gaya gravitasi dan gaya antar partikel. Dalam pengaplikasiannya, *fluidized bed reaktor* pada umumnya tidak hanya menggunakan satu jenis partikel melainkan menggunakan lebih dari satu jenis partikel. Penggunaan lebih dari satu jenis partikel dalam *bed* menyebabkan munculnya fenomena pencampuran maupun pemisahan partikel ketika *fluidized bed reactor* beroperasi.<sup>[5]</sup>



**Gambar 2.6** *Fluidized Bed Reactor*

## 2.7 Storage Tank

*Storage Tank* atau tangki pada dasarnya dipakai sebagai tempat penyimpanan material baik berupa benda padat, cair, maupun gas (D Cendana, 2011). Fungsi serta pengaplikasiannya sangatlah banyak, seperti halnya untuk tangki uji berguna untuk melakukan uji laju reduksi produksi suatu sumur. Biasanya berukuran lebih kecil dan diletakkan diatas permukaan tanah. Tangki produksi berguna sebagai tangki untuk menampung hasil produksi. Ditinjau dari bahan pembuatnya ada 3 jenis tangki air yaitu dari bahan plastik PE (*Poly Etilene*), dari bahan *stainless steel* dan dari bahan fiber.

Tangki dengan material dari bahan plastik PE (*Poly Etilene*) adalah tangki penampung yang digunakan dimana bagian dalamnya dilengkapi dengan lapisan anti lumut sehingga menjaga kualitas air yang disimpan didalamnya.<sup>[6]</sup>





**Gambar 2.7** Tangki Penampung Air

### **2.8 Ball Valve**

*Control Valve* adalah suatu jenis pengendali akhir atau *final control element* yang paling umum digunakan untuk memanipulasi proses laju aliran fluida. Suatu proses aliran fluida pada *control valve* bekerja tidak hanya pada posisi menutup secara penuh (*fully closed*) atau membuka secara penuh (*fully opened*) tetapi dapat juga dikendalikan melalui *manipulated variable* atau input dari suatu proses yang dapat dimanipulasi atau diubah-ubah untuk mengatur besaran bukaan *valve* agar proses variabel selalu sama dengan *set point* yang dikehendaki. Contoh jenis dari *control valve* adalah *manual valve*. *Manual valve* adalah katup-katup yang terpasang di saluran fluida gas atau cair yang cara pengoperasiannya masih didominasi oleh manusia yaitu pada saat membuka atau menutup aliran fluidanya. Tipe katup yang dioperasikan secara manual banyak ragamnya, antara lain : *ball*, *globe*, *butterfly*, *gate* dan lain-lain. Adapun alat untuk membuka/menutup katup secara manual yaitu tipe : *level/tungkai*, *hand wheel*/berbentuk steer mobil yang terdapat pada katup tersebut.<sup>[7]</sup>



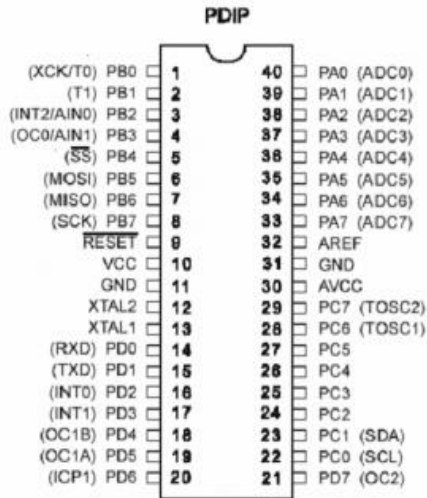
**Gambar 2.8** *Ball Valve*

## **2.9 Mikrokontroler Atmega32**

Mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus. Sederhananya, cara kerja mikrokontroler sebenarnya hanya membaca dan menulis data. Mikrokontroler merupakan komputer didalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Microcontroller AVR (Alf and Vegard's Risc processor) standar memiliki arsitektur 8bit, dimana semua instruksi dikemas dalam kode 16-bit, dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus clock. AVR berteknologi RISC (*Reduced Instruction Set Computing*). AVR dapat dikelompokkan menjadi empat 6 kelas, yaitu keluarga ATtiny, keluarga AT86RFxx, keluarga ATmega, dan AT86RFxx. Pada dasarnya, yang membedakan masing-masing kelas adalah memori. Berikut adalah fitur yang terdapat pada *microcontroller* ATmega32 yaitu:

1. Saluran I/O ada 32 buah, yaitu Port A, Port B, Port C, dan Port D.
2. ADC (*Analog to Digital Converter*) 10 bit sebanyak 8 *channel*.
3. Tiga buah *Timer/Counter*.
4. CPU yang terdiri dari 32 buah *register*.
5. Memiliki 131 instruksi yang membutuhkan 1 siklus *clock*.
6. *Watchdog timer* dengan osilator internal.
7. Dua buah *timer/counter* 8 bit, satu *buah timer/counter* 16 bit.
8. Tegangan operasi 2.7 V – 5.5 V pada ATmega 16L.

9. Internal SRAM sebesar 1 KB.
10. Memori *flash* sebesar 32KB dengan kemampuan *Read While Write*.
11. Unit interupsi internal dan eksternal.
12. Port antarmuka SPI.
13. EEPROM sebesar 512 *byte* yang dapat diprogram saat operasi.
14. Antarmuka komparator analog.
15. 4 channel PWM.
16. 2x8 *general purpose register*.
17. Hampir mencapai 16 MIPS pada Kristal 16 MHz.
18. Port USART yang dapat diprogram untuk komunikasi serial.



**Gambar 2.10** Konfigurasi Pin *Mikrokontroler* Atmega 32

Fungsi umum dari susunan pin *microcontroller* ATmega32 adalah sebagai berikut:

1. VCC merupakan catu daya positif.
2. GND sebagai pin *ground* catu daya negatif. 8
3. Port A (PA0..PA7) merupakan pin I/O dua arah dan dapat diprogram sebagai pin dari ADC.
4. Port B (PB0..PB7) merupakan pin I/O dua arah dan fungsinya sebagai, *timer/counter*, komparator analog, dan SPI.

5. Port C (PC0..PC7) merupakan pin I/O dua arah dan fungsinya sebagai, TWI, komparator analog, dan *timer* osilator.
6. Port D (PD0..PD7) merupakan pin I/O dua arah dan fungsinya sebagai, komparator analog, interupsi eksternal dan komunikasi serial.
7. *Reset* merupakan pin yang digunakan untuk me-*reset microcontroller*.
8. XTAL 1 dan XTAL 2 sebagai pin *clock* eksternal. Pada *microcontroller* membutuhkan sumber (*clock*) agar dapat mengeksekusi instruksi yang ada di memori. Semakin tinggi nilai kristalnya, maka semakin cepat *microcontroller* tersebut.
9. AVCC sebagai pin tegangan untuk ADC.
10. AREF sebagai pin tegangan referensi.<sup>[8]</sup>

### **2.10 Liquid Crystal Display (LCD)**

*LCD (Liquid Crystal Display)* adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. Dipasaran tampilan LCD sudah tersedia dalam bentuk modul yaitu tampilan LCD beserta rangkaian pendukungnya. LCD mempunyai pin data, kontrol catu daya, dan pengatur kontras tampilan.

LCD juga merupakan perangkat *display* yang paling umum dipasangkan di *Mikrokontroller*, mengingat ukurannya yang kecil dan kemampuannya menampilkan karakter atau grafik yang lebih dibandingkan display *seven-segmen*. Pada pengembangan sistem *embedded*, LCD mutlak diperlukan sebagai sumber pemberi informasi utama, misalnya alat pengukur kadar gula darah, penampil waktu jam, penampil *counter* putaran motor industri dan lain-lain.



**Gambar 2.11** *Liquid Crystal Display*

Sebagai berikut adalah modul LCD 20x4 karakter yang akan digunakan dalam final project ini. Salah satu alasan mengapa modul LCD dipakai dalam proyek akhir ini adalah untuk menunjukkan angka pengukuran pada lux meter digital. Dengan mikrokontroler kita dapat mengendalikan suatu peralatan agar dapat bekerja secara otomatis. Untuk mengakses LCD 20x4 harus melakukan konfigurasi pin dari LCD dengan pin I/O mikrokontroler tersebut.<sup>[9]</sup>

Berikut tabel deskripsi pin pada LCD :

No	<i>Symbol</i>	<i>Function</i>
1	VSS	<i>GND pin, 0V</i>
2	VDD	<i>Positive power pin, +5V</i>
3	V0	<i>LCD drive voltage input pin</i>
4	RS	<i>Data/Instruction select input pin</i>
5	R/W	<i>Read/Write select input pin</i>
6	E	<i>Enable input pin</i>
7-14	D0-D7	<i>Data bus line</i>
15	Led A	<i>Led power supply</i>
16	Led K	<i>Led power supply</i>

### **2.11 Heater**

*Electrical Heating Element* (elemen pemanas listrik) banyak dipakai dalam kehidupan sehari-hari, baik didalam rumah tangga ataupun peralatan dan mesin industri. Elemen pemanas merupakan alat yang berfungsi sebagai salah satu kegiatan kerja untuk mendapatkan suhu dari suhu rendah suatu zat sampai ke suhu

tinggi. Sebagai sumber panas yang dihasilkan oleh elemen pemanas listrik ini bersumber dari kawat ataupun pita bertahanan listrik tinggi (*Resistance Wire*) biasanya bahan yang digunakan adalah kawat niklin yang digulung menyerupai bentuk spiral dan dimasukkan dalam selongsong/pipa sebagai pelindung, kemudian dialiri arus listrik pada kedua ujungnya dan dilapisi oleh isolator listrik yang mampu meneruskan panas dengan baik hingga aman jika digunakan. Bentuk dan tipe dari *Electrical Heating Element* ini bermacam macam disesuaikan dengan fungsi, tempat pemasangan dan media yang akan di panaskan.

Persyaratan elemen pemanas antara lain :

- Harus tahan lama pada suhu yang dikehendaki.
- Sifat mekanisnya harus kuat pada suhu yang dikehendaki.
- Koefisien muai harus kecil, sehingga perubahan bentuknya pada suhu yang dikehendaki tidak terlalu besar.
- Tahanan jenisnya harus tinggi.
- Koefisien suhunya harus kecil, sehingga arus kerjanya sedapat mungkin konstan.

*Chromalox Heater* atau lebih sering kita kenal dengan *water heater/ pemanas air*, merupakan jenis *heater* celup yang aplikasinya digunakan untuk air ukuran standar dengan diameter pipa 8mm dengan daya sebesar 230V 1000W, 1500W, 2000W, 3000W, 4000W, 5000W, 6000W.<sup>[10]</sup>



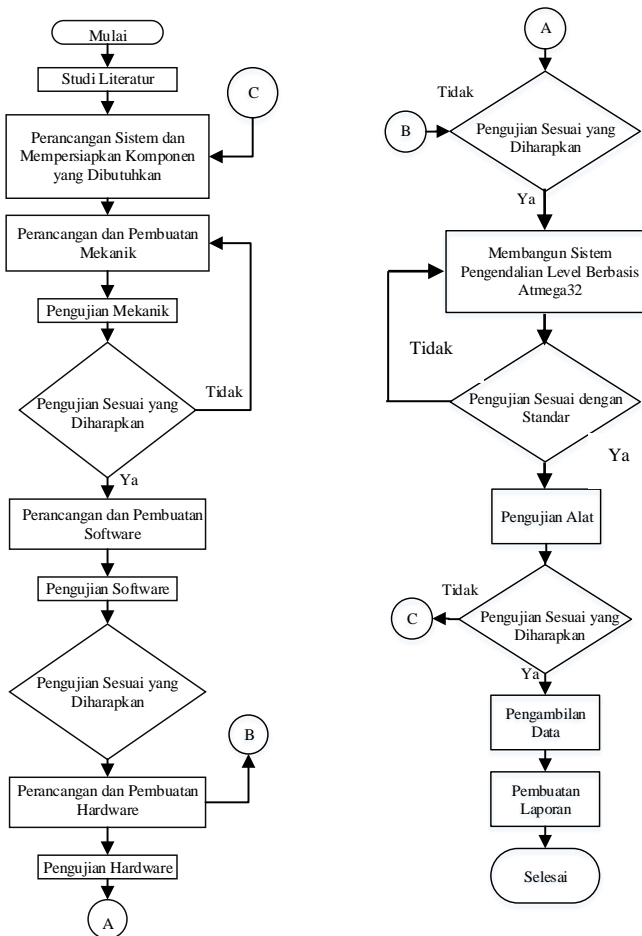
**Gambar 2.12** *Chromalox Heater*

*Halaman ini sengaja dikosongkan*

## BAB III PERANCANGAN DAN PEMBUATAN ALAT

### 3.1 Diagram Alir Perancangan dan Pembuatan Alat

Langkah-langkah dalam tugas akhir ini digambarkan dalam diagram alir pada Gambar 3.1 berikut ini :



**Gambar 3.1** Diagram Alir Tugas Akhir



Dengan adanya skema diagram alir Tugas Akhir mengenai “Rancang Bangun Sistem Pengendalian *Level* pada Tangki Pemanas *Plant Fluidized Bed Reactor* Pengolahan *Oil Sludge*” yang telah digambarkan, maka adapun penjelasan pada setiap langkah-langkah yang telah digambarkan adalah sebagai berikut:

### **3.2 Studi Literatur**

Tahap awal pada skema diagram alir Tugas Akhir ini dimulai dengan adanya studi literatur sebagai upaya pemahaman terhadap materi yang menunjang tugas akhir mengenai “Rancang Bangun Sistem Pengendalian *Level* pada Tangki Pemanas *Plant Fluidized Bed Reactor* Pengolahan *Oil Sludge*”. Studi literatur ini dilakukan dengan mencari dan mempelajari informasi dari *e-book* maupun buku manual mengenai uji performansi suatu sensor yang diaplikasikan pada Tugas Akhir ini.

### **3.3 Perancangan Sistem dan Pembuatan Alat**

Setelah melakukan studi literatur, selanjutnya adalah melakukan perancangan sistem dan pembuatan rancang bangun reaktor yang terdiri dari perancangan pembuatan *hardware* dan *software*, serta pembuatan mekanik. *Hardware* dan *software* yang telah dibuat kemudian diintegrasikan melalui mikrokontroler ATmega 32. Selanjutnya diintegrasikan dengan mekanik rancang bangun reaktor agar dapat mengetahui nilai *level* yang terukur.

#### **3.3.1 Pembuatan Mekanik**

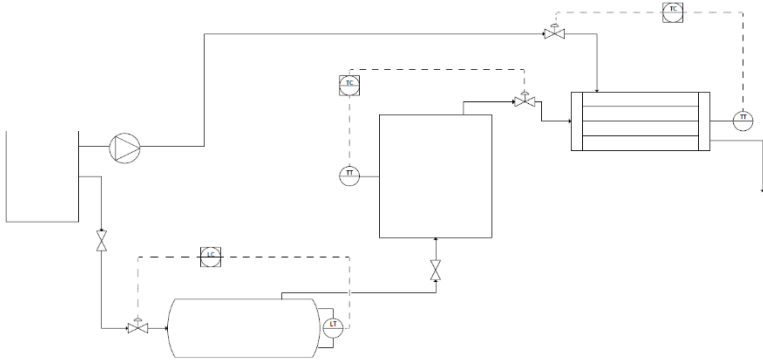
Pembuatan mekanik meliputi pembuatan sistem pada rancang bangun *fluidized bed reactor* dimana terdapat 3 sensor yaitu *level switch* dan 2 *thermocouple* tipe K, terdapat 3 aktuator yaitu 3 motor servo yang saling terintegrasi, tangki pemanas, reaktor dan kondensor beserta kerangka, pemasangan panel kontrol dan terdapat penampungan air berupa 2 buah bak air. Dari desain rancang bangun diatas dapat diketahui spesifikasi dari masing-masing komponen, berikut adalah tabel spesifikasinya:

**Tabel 3.1** Tabel Spesifikasi Rancang Bangun

<b>Kode</b>	<b>Keterangan</b>	<b>Spesifikasi</b>
A	Tangki Pemanas	Diameter 20 cm, panjang 26 cm
B	Reaktor	Diameter 26 cm, panjang 38 cm
C	Kondensor	Diameter 7,5 cm, panjang 70 cm
D	<i>Level switch</i>	Dengan kondisi <i>high</i> dan <i>low</i>
E	<i>Thermocouple Type K</i>	Dengan spesifikasi range suhu 0-400°C
F	Motor Servo MG996R	Dengan spesifikasi torsi 11 kgf/cm
G	<i>Heater MT</i>	Daya sebesar 3500 Watt

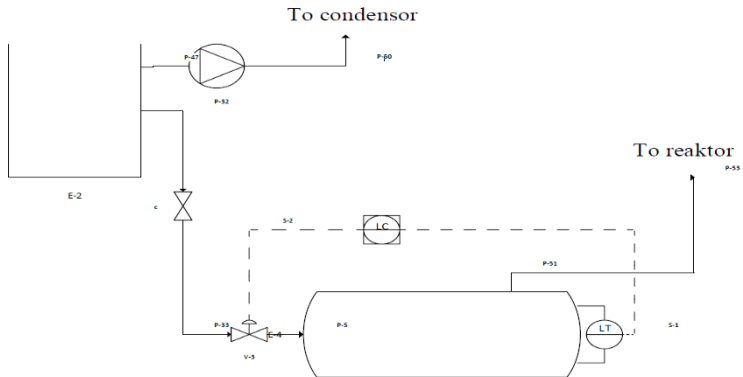
Pada gambar 3.2 merupakan gambar P&ID dari *plant fluidized bed reactor*. Terdapat 3 sistem kontrol pada *plant* tersebut. Ketiga sistem kontrol tersebut meliputi 1 sistem pengendalian *level* dan 2 sistem pengendalian temperatur. Proses dimulai dengan fluida dialirkan ke dalam tangki pemanas. Saat ketinggian fluida dalam tangki pemanas mencapai *set point*, *valve* akan menutup sedangkan pada saat *level* kurang dari *set point*, *valve* akan membuka. Setelah terbentuk uap yang diinginkan, uap akan mengalir melalui *valve* menuju reaktor. Uap tersebut masuk ke dalam reaktor untuk proses pemisahan *sludge* dengan cara menguapkan *sludge*. Saat suhu dalam reaktor kurang dari *set point*, *valve* akan menutup, saat suhu mencapai *set point*, *valve* akan membuka yang kemudian *sludge* menuju kondensor untuk dikondensasikan. *Sludge* hasil penguapan dicairkan menjadi

minyak dengan cara merubah fase *sludge* dari lumpur menjadi cairan. Saat suhu dalam kondensor kurang dari *set point*, *valve* akan menutup, saat suhu mencapai *set point*, *valve* akan membuka, kemudian *sludge* yang sudah berubah fase menjadi cair berupa minyak murni ditampung ke tangki hasil penampungan.



**Gambar 3.2** P&ID *plant fluidized bed reactor*

Pada gambar 3.3 merupakan gambar P&ID dari sistem pengendalian *level* yang ada pada *plant fluidized bed reactor*.



**Gambar 3.3** P&ID Sistem Pengendalian *Level*

Pada Gambar 3.4 dibawah ditunjukkan gambar keseluruhan plant *Fluidized Bed Reactor*. Pada *plant* tersebut terjadi proses pemisahan *oil sludge* dengan uap air dengan memanaskan *sludge* hingga kandungan minyak yang tercampur dengan *sludge* ikut mengalir ke atas bersama *steam* sehingga mendapatkan minyak yang diinginkan.



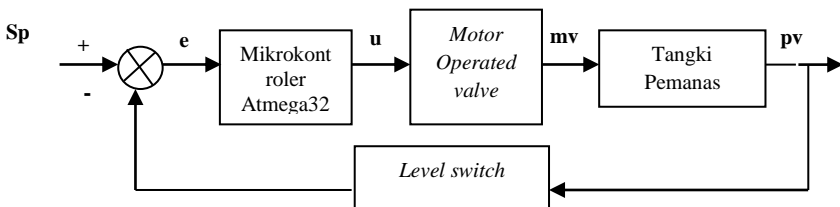
**Gambar 3.4** *Plant Fluidized Bed Reactor*

Pada gambar 3.5 dibawah merupakan tangki pemanas dari *plant fluidized bed reactor*. Terdapat beberapa komponen sistem pengendalian *level*. Komponen tersebut berupa sensor *level switch*, tangki penampung air, tangki pemanas serta motor servo. Penempatan sensor *level switch* berada di atas tangki pemanas untuk mendeteksi ketinggian fluida. Kondisi air yang akan dipanaskan harus dalam keadaan konstan dan *heater* pemanas air harus selalu kontak

dengan air sehingga pengendalian *level* diperlukan agar *level* air selalu konstan. *Level* air yang terlalu tinggi akan menyebabkan *steam* yang dihasilkan akan mengandung uap air sehingga akan membahayakan proses berikutnya. Sebaliknya bila *level* air terlalu rendah maka akan merusak komponen pemanas air yang berupa *heater*.



**Gambar 3.5** Tangki Pemanas



**Gambar 3.6** Diagram Blok Pengendalian *Level*

Gambar 3.6 merupakan diagram blok sistem pengendalian *level* pada reaktor *plant fluidized bed reactor*. Mikrokontroler ATmega 32 sebagai kontroler. Motor servo sebagai pengendali akhir atau aktuator, sedangkan tangki pemanas sebagai tempat proses. Sensor yang digunakan pada pengendalian *level* tersebut adalah sensor *Level switch*. SP (*Set Point*) merupakan besar proses variabel yang dikehendaki. *Error* merupakan proses mengurangi besaran *set point* dengan nilai pengukuran. MV (*Manipulated Variable*) merupakan input dari suatu proses yang dapat dimanipulasi agar besarnya variabel proses sama dengan *set point*. PV (*Process Variable*) merupakan suatu proses yang dikendalikan pada suatu sistem.

Uap yang dihasilkan dari tangki pemanas akan dialirkan menuju reaktor akan dideteksi suhunya oleh sensor termokopel. Kemudian sensor termokopel akan mengirimkan data ke mikrokontroler ATmega 32. Lalu mikrokontroler mengirimkan ke LCD untuk menampilkan display *level* air pada tangki pemanas. Digunakan  $set\ point \geq 10\ cm$  untuk membuka *valve*. Jadi, ketika *level* air pada tangki pemanas mencapai 10 cm atau lebih, *valve* akan menutup. Dan apabila *level* air kurang dari 10 cm, *valve* akan membuka.

### **3.3.2 Perancangan *Hardware* dan *Software***

Untuk perancangan *hardware* dimulai dengan integrasi sensor, membuat rangkaian LCD dan integrasi motor servo. Sedangkan untuk perancangan *software* melakukan program pada software CodeVision AVR.

Sistem pengendalian *level* pada tangki pemanas ini menggunakan sensor *level switch*. Sensor *level switch* tersebut berfungsi untuk mendeteksi *level* air yang ada pada tangki pemanas. Pembuatan *hardware* dimulai dengan mengintegrasikan sensor *level switch* ke mikrokontroler ATmega 32. Mikrokontroler ATmega 32 berfungsi sebagai kontroler untuk sistem pengendalian *level* ini.

a. Perancangan *Hardware*

Pada gambar 3.7 merupakan konfigurasi dari sensor *Level switch* yang telah terintegrasi dengan mikrokontroler ATmega 32.



**Gambar 3.7** Konfigurasi Sensor *Level Switch*

Pada gambar 3.8 merupakan konfigurasi aktuator motor servo dengan mikrokontroler ATmega 32 dan *power supply* 12V DC. Output dari mikrokontroler ATmega 32 sebesar 5V DC. Pin data motor servo dihubungkan dengan mikrokontroler ATmega 32 dan pin *ground* serta vcc motor servo dihubungkan pada *power supply* 12VDC. Setelah terselesaikannya konfigurasi dari aktuator, kemudian mengintegrasikan semua komponen.



**Gambar 3.8** Konfigurasi Motor Servo

Pada gambar 3.9 merupakan konfigurasi rangkaian LCD 20x4. LCD berfungsi untuk menampilkan suatu karakter huruf maupun angka. LCD 20x4 mempunyai tampilan 20 kolom dan 4 baris. Perancangan rangkaian LCD digunakan untuk menampilkan nilai ketinggian fluida pada reaktor. Digunakan mikrokontroler ATmega 32 untuk mengkoneksikan LCD 20x4.



**Gambar 3.9** Konfigurasi *display* LCD

b. Perancangan *Software*

Pada pembuatan tugas akhir ini digunakan suatu *software* guna penunjang daripada sistem kontrol. *Software* yang digunakan untuk membuat program pada mikrokontroler ATmega 32 adalah *CodeVision AVR*. *listing program* atau koding dibuat untuk mengolah data yang telah dibaca oleh sensor untuk ditampilkan ke LCD. Gambar 3.10 ini merupakan cuplikan dari *listing program* untuk instruksi mikrokontroler Atmega32 pada tugas akhir ini.



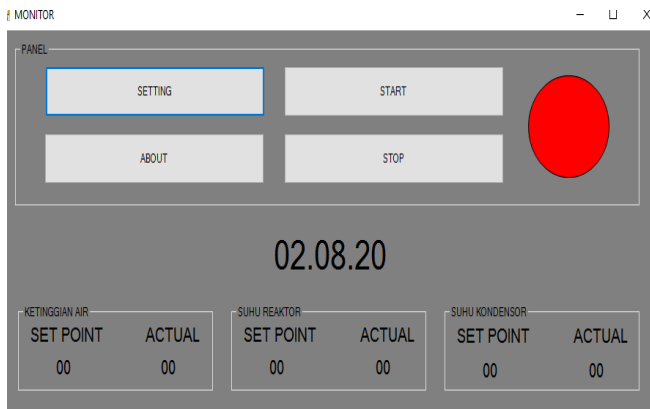
```

15
16 Chip type           : ATmega32
17 Program type       : Application
18 AVR Core Clock frequency: 8.000000 MHz
19 Memory model      : Small
20 External RAM size  : 0
21 Data Stack size   : 512
22 *****/
23
24 #include <mega32.h>
25 #include <delay.h>
26 #include <stdlib.h>
27
28 // Alphanumeric LCD functions
29 #include <alcd.h>
30
31 #define thermocouple1 PORTA.0
32 #define thermocouple2 PORTA.1
33
34 #define trigger_ultra PORTA.7
35 #define echo_ultra    PINA.6
36
37 #define ButtonUp      PINA.2
38 #define ButtonDown    PINA.3

```

**Gambar 3.10** Listing Program Atmega 32 pada CVAVR

Setelah program yang dibuat pada software CodeVision AVR sudah sesuai dengan aksi yang terjadi pada plant, sistem pengendalian *level* ini kemudian dibuat tampilan data di sebuah Personal Computer (PC). Selain itu data disimpan secara real time. Software yang digunakan untuk tampilan serta penyimpanan data tersebut adalah Visual Studio. Data yang ditampilkan pada software tersebut adalah nilai dari ketinggian fluida pada tangki pemanas beserta grafik respon yang dihasilkan dari sistem pengendalian *level*.



**Gambar 3.11** Tampilan pada Visual Studio

### 3.4 Integrasi

Pada tahap ini dilakukan proses *download* program CVAVR ke *minimum system* ATmega 32 dengan *software* Khazama AVR *programmer*, kemudian *minimum system* yang telah berisi program dihubungkan dengan rangkaian *sensing element* dan rangkaian modul LCD.

### 3.5 Pengujian Alat

Pada rancang bangun *fluidized bed reactor* dilakukan percobaan sederhana yaitu dengan menyalakan *heater* untuk memanaskan air yang uapnya digunakan memanaskan *oil sludge* di dalam reaktor dengan kondisi valve adalah *normally close* sehingga uap dapat terkumpul didalam reaktor. Sistem pengendalian suhu pada tangki pemanas diuji coba menggunakan sensor *thermocouple* yang berfungsi untuk mendeteksi suhu di dalam reaktor, apabila suhu sudah mencapai *set point* maka aktuator berupa motor servo akan aktif untuk membuka *valve*. Kemudian uap yang sudah terbentuk akan mengalir ke atas menuju kondensor. Apabila aksi dari aktuator tidak sesuai dengan *set point*, maka dilakukan sebuah *troubleshooting* untuk mencari penyebab dari tidak berjalannya suatu proses pengendalian

tersebut. Apabila semua rancang bangun pada reaktor dapat bekerja dengan baik, maka selanjutnya dapat dilakukan pengambilan data.

### 3.6 Pengambilan Data

Pada tahap pengambilan data untuk karakteristik statik dan kalibrasi merupakan tahap dimana melihat spesifikasi yang dimiliki oleh sistem pengendalian yang telah dibuat, dengan adanya data tersebut dapat diketahui performansi sistem pengendalian. Pada karakteristik statik alat yang dicari yaitu nilai *range*, *span*, resolusi, sensitivitas, *non-linieritas*, *hysteresis*, serta akurasi. Sedangkan untuk data kalibrasi digunakan untuk mencari nilai ketidakpastian dari hasil pengukuran ketika menggunakan perangkat sistem pengendalian untuk arus dan tegangan *input-output* ini. Berikut merupakan langkah-langkah pengambilan data karakteristik statik:

- Nilai *range*, *span*, dan resolusi sistem pengendalian dicatat sesuai spesifikasi sensor arus dan tegangannya.
- Nilai sensitivitas ditentukan dengan rumus:

$$\text{Sensitivitas} = \frac{\Delta O}{\Delta I}$$

- Nilai *linieritas* ditentukan dengan rumus:

$$O - O_{min} = \left[ \frac{O_{maks} - O_{min}}{I_{maks} - I_{min}} \right] (I - I_{min})$$

$$O_{ideal} = KI + \alpha$$

Dimana:

$$K = \text{Kemiringan garis lurus ideal} = \frac{O_{maks} - O_{min}}{I_{maks} - I_{min}}$$

$$\alpha = \text{Koefisien garis lurus ideal} = O_{min} - KI_{min}$$

- Histerisis ditentukan dengan melakukan pengambilan data input naik dan turun, dengan persamaan histerisis:

$$H(I) = O(I)_{I\downarrow} - O(I)_{I\uparrow}$$

$$\% \text{ Maksimum Histerisis} = \frac{H}{O_{maks} - O_{min}} \times 100\%$$

- Nilai akurasi ditentukan dari pembacaan alat dengan pembacaan standar, nilai akurasi ditentukan dari nilai kesalahan akurasi dengan persamaannya, yaitu:

$$A = 1 - \left| \text{Rata - Rata} \frac{(\text{Pemb. std} - \text{Pemb. alat})}{\text{Pemb. std}} \right|$$

### 3.7 Analisa Data dan Pembahasan

Setelah pembuatan rancangan telah selesai dengan hasil yang sesuai, kemudian dilakukan analisa data dengan memanfaatkan hasil dari uji performansi dan sistem pengendalian.

### 3.8 Penulisan Laporan

Setelah semua hasil yang diinginkan tercapai, kemudian semua hasil mulai dari studi literatur sampai dengan analisa data dan kesimpulan dicantumkan dalam sebuah laporan.

*Halaman ini sengaja dikosongkan*

## **BAB IV**

### **PENGUJIAN ALAT DAN PEMBAHASAN**

Setelah perancangan sistem pengendalian *level* pada tangki pemanas *plant fluidized bed reactor* dibuat maka pengujian baik dari *hardware*, *controlling*, dan *software* perlu dilakukan. Pengujian dilakukan untuk mendapatkan data-data dari alat yang dirancang guna mengetahui spesifikasi serta performansi dari alat secara keseluruhan dan seberapa besar *error* atau kesalahan yang terjadi pada alat berdasarkan respon sistem dari nilai set poin yang diberikan.

Secara mekanisme kerja dari perancangan sistem pengendalian *level* pada tangki pemanas *plant fluidized bed reactor* ini menjaga ketinggian air pada tangki pemanas karena kondisi air yang akan dipanaskan harus dalam keadaan konstan dan *heater* pemanas air harus selalu kontak dengan air. Pada pengendalian *level* tangki pemanas, sensor *level switch* digunakan untuk mengukur ketinggian air pada tangki pemanas. Hasil koreksi, pengukuran, serta perhitungan oleh *kontroler* akan memberi perintah berupa eksekusi bagi motor servo sebagai aktuator untuk membuka atau menutup *valve* dalam sistem pengendalian *level*.

Berikut merupakan hasil perancangan sistem pengendalian *level* pada tangki pemanas *plant fluidized bed reactor* berbasis *microcontroller* ATmega32.



**Gambar 4.1** *Plant fluidized bed reactor*

Pada proses pengendalian *level* air pada tangki pemanas ini digunakan 1 buah sensor. Sensor yang digunakan pada proses pengendalian ini adalah sensor *level switch* dimana sensor ini mempunyai dua kondisi yaitu *high* dan *low*.

#### **4.1 Pengujian Sensor *Level Switch***

Untuk mengetahui keakuratan dari sensor *level switch* yang digunakan pada sistem, perlu adanya sebuah uji pada sensor tersebut. Pengujian dilakukan dengan menggunakan meteran sebagai acuan standar dengan satuan cm.

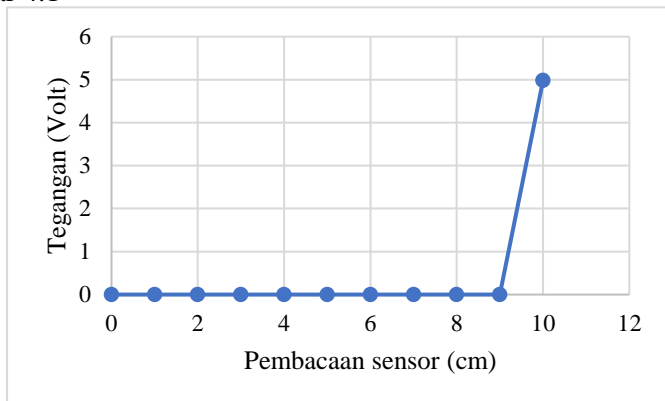
**Tabel 4.1** Hasil Pengujian Sensor *Level Switch* Saat Kondisi *High*

No.	<i>Level</i> (cm)	Status
1.	0	<i>Off</i>
2.	1	<i>Off</i>
3.	2	<i>Off</i>
4.	3	<i>Off</i>

5.	4	<i>Off</i>
6.	5	<i>Off</i>
7.	6	<i>Off</i>
8.	7	<i>Off</i>
9.	8	<i>Off</i>
10	9	<i>Off</i>
11.	10	<i>Off</i>

Dari Tabel 4.1 diatas dapat diketahui perubahan tiap keluaran nilai tegangan *level switch* ketika kondisi *low* dan *high*. Ketika *level switch* pada kondisi *low* atau belum memenuhi *set point*, maka keluaran nilai voltasenya sebesar 0 volt. Ketika *level switch* pada kondisi *high* atau telah mencapai *set point*, maka keluaran nilai voltase sebesar 5 volt sesuai dengan besar sumber yang digunakan.

Dari tabel tersebut diperoleh grafik hubungan antara pengaruh nilai tegangan terhadap pembacaan *level set point* sensor *high*. Adapun grafik tersebut telah ditunjukkan pada gambar 4.1



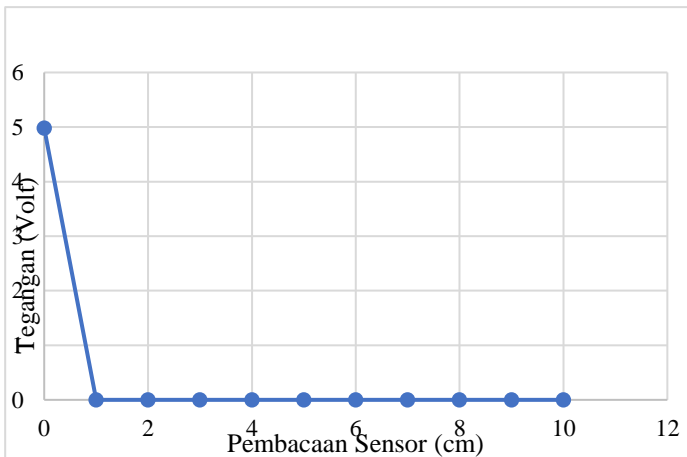
**Gambar 4.2** Grafik Tegangan Saat Kondisi *High*



**Tabel 4.2** Hasil Pengujian Sensor *Level Switch* Saat Kondisi *Low*

No.	Level (cm)	Status
1.	0	<i>On</i>
2.	1	<i>Off</i>
3.	2	<i>Off</i>
4.	3	<i>Off</i>
5.	4	<i>Off</i>
6.	5	<i>Off</i>
7.	6	<i>Off</i>
8.	7	<i>Off</i>
9.	8	<i>Off</i>
10.	9	<i>Off</i>
11.	10	<i>Off</i>

Dari tabel tersebut diperoleh grafik hubungan antara pengaruh nilai tegangan terhadap pembacaan *level set point* sensor *low level switch*. Adapun grafik tersebut telah ditunjukkan pada gambar 4.2

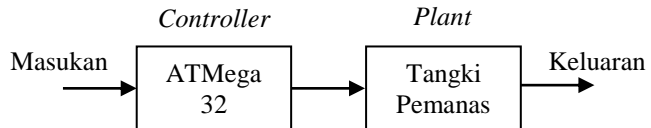
**Gambar 4.3** Grafik Tegangan Saat Kondisi *Low*

## 4.2 Respon sistem

Setelah plant tangki pemanas berjalan dengan baik, perlu dilakukan uji respon. Uji respon dilakukan untuk mengetahui berapa selang waktu yang dibutuhkan suatu sistem pengendalian untuk mencapai *set point*.

### 4.2.1 *Open loop*

Sistem control dimana keluaran tidak memberikan efek terhadap besaran masukan. Sensor *level switch* mendeteksi adanya *level* pada tangki pemanas dalam satuan cm yang nantinya dapat menggerakkan motor servo untuk membuka *valve*.



**Gambar 4.4** Sistem Pengendalian *Level* Terbuka pada Tangki Pemanas

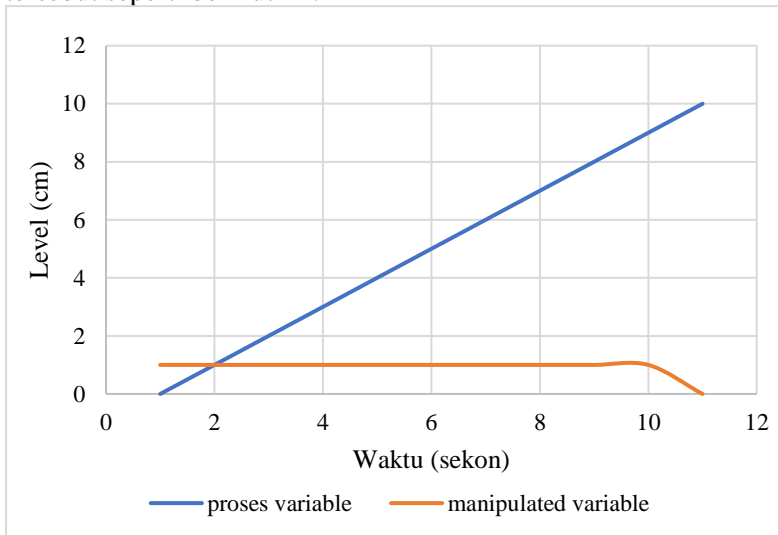
Berikut merupakan tabel hasil pengujian sistem pengendalian *level open loop*.

**Tabel 4.3** Hasil Pengujian *Open loop*

Waktu (s)	Pembacaan <i>level</i> (cm)
0	0
4,15	1
9,23	2
13,46	3
17,82	4
21,13	5

27,77	6
32,56	7
37,65	8
45,22	9
49,11	10

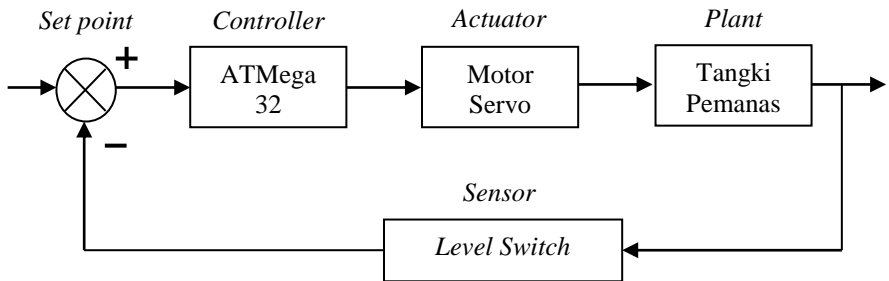
Dari tabel tersebut, diperoleh grafik hubungan antara pembacaan *level* permukaan air dengan waktu. Adapun grafik tersebut seperti berikut ini:



**Gambar 4. 5** Grafik *Open Loop* Sistem Pengendalian *Level*

#### 4.2.2 *Close Loop*

Sistem *control* dimana besaran keluaran memberikan efek terhadap besaran masukan sehingga besaran yang dikontrol dapat dibandingkan terhadap nilai yang diinginkan.



**Gambar 4.6** Diagram Blok Sistem Pengendalian *Level*

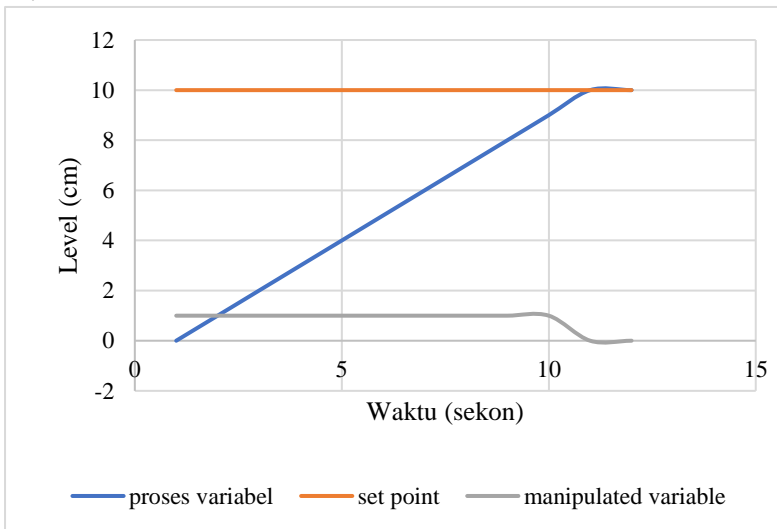
Gambar 4.6 diatas merupakan gambar diagram blok sistem pengendalian *level* pada tangki pemanas yang terdiri dari mikrokontroler ATMega 32 sebagai *controller*, motor servo sebagai *actuator*, tangki pemanas sebagai tempat pemrosesan, dan sensor *level switch* sebagai sensor untuk sensing *level* didalam tangki pemanas (dalam satuan cm).

**Tabel 4.4** Hasil Pengujian *Close Loop*

<b>Waktu (s)</b>	<b>Set point (cm)</b>	<b>Pembacaan level (cm)</b>	<b>Bukaan Valve</b>
0	10	0	100%
01,18	10	1	100%
03,57	10	2	100%
06,40	10	3	100%
09,00	10	4	100%
10,86	10	5	100%
13,24	10	6	100%
15,62	10	7	100%
17,33	10	8	100%

20,45	10	9	100%
22,52	10	10	0%
85,22	10	10	0%

Dari tabel tersebut dengan *set point* sebesar 10 cm, diperoleh grafik hubungan antara pembacaan *level* ketinggian permukaan air dengan *level set point*. Adapun grafik tersebut seperti berikut ini:



**Gambar 4.7** Grafik *Closed Loop* Sistem Pengendalian *Level*

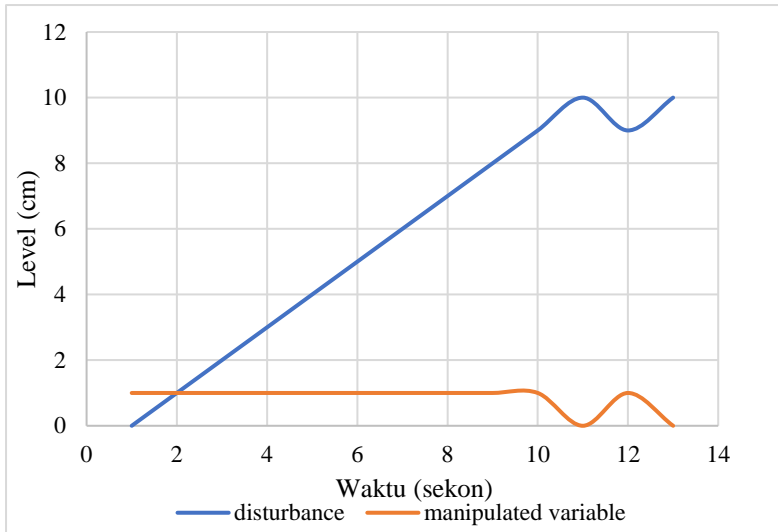
### 4.3 Uji *Disturbance*

Uji performansi yang selanjutnya diterapkan pada sistem berupa uji *disturbance* dengan mengubah laju aliran pada outlet tangki pemanas untuk melihat performansi pengendali tersebut dalam mempertahankan *set point* terdapat perubahan *disturbance*.

**Tabel 4.5** Hasil Pengujian *Disturbance*

<b>Waktu (s)</b>	<b>Pembacaan <i>level</i> (cm)</b>
0	0
01,18	1
03,57	2
06,40	3
09,00	4
10,86	5
13,24	6
15,62	7
17,33	8
20,45	9
22,52	10
82,13	9
85,22	10
150,34	9
153,42	10

Uji *disturbance* yang dilakukan dengan mengubah laju aliran pada outlet tangki pemanas dan didapatkan hasil respon perubahan *level* yang dapat dilihat pada gambar 4.8 berikut



**Gambar 4.8** Uji *Disturbance*

#### 4.4 Pembahasan

Pada tugas akhir ini telah dirancang sistem pengendalian *level* pada tangki pemanas *plant fluidized bed reactor*. Sistem pengendalian *level* tersebut untuk menjaga ketinggian air pada tangki pemanas. Sensor yang digunakan adalah sensor *level switch*. Menggunakan *mikrokontroler Atmega32* sebagai kontroler. Sistem pengendalian tersebut memberikan aksi pada motor servo untuk membuka dan menutup *valve*. Digunakan *set point* 10 cm untuk menutup *valve*.

Pengujian sensor dilakukan untuk mengetahui karakteristik dari sensor tersebut dan telah diperoleh tiga data mengenai pembacaan dari hasil pengujian sensor *level switch*, hasil pengujian sistem pengendalian *level* air dan uji *disturbance*. Sensor ketinggian *level switch* yang telah dirancang sedemikian rupa guna mengetahui *level* permukaan air pada tangki pemanas ini memiliki keluaran data sinyal diskrit atau digital 10 bit. Sinyal keluaran dari *level switch* yang berupa tegangan dihasilkan

dengan besaran 0 volt atau 5 volt sesuai dengan sumbernya sehingga dengan mudah dapat dibaca sebagai data *digital* bagi kontroler.

Selanjutnya saat *level* permukaan berada pada ketinggian 10 cm, berarti *level* telah mencapai nilai *set point* dan sensor berada pada kondisi *high*. Saat *level* mencapai kurang dari 10 cm, berarti *level* belum mencapai nilai *set point* dan sensor berada pada kondisi *low*.

Adapun hasil respon sistem pada pengujian *open loop* dan *close loop*, yaitu pengujian *open loop* dilakukan untuk melihat karakteristik dari proses dengan melihat respon perubahan *level* pada tangki pemanas terhadap waktu tanpa adanya *controler* dari proses pengendalian. Dari proses tersebut dapat dilihat bahwa proses tersebut memerlukan *controler* untuk menjamin agar proses dapat berjalan dengan baik. Pengujian *closed loop* dilakukan untuk melihat respon sistem dari tangki pemanas. Dari proses tersebut dapat diketahui bahwa sistem dari tangki pemanas membutuhkan pengendalian agar dapat menjaga *set point*. Kemudian hasil dari uji *disturbance* yaitu dengan mengubah laju aliran pada outlet tangki pemanas. Uji *disturbance* dilakukan untuk melihat performansi pengendali tersebut dalam mempertahankan *set point* terhadap perubahan.

Adapun respon dari motor servo sebagai aktuatur diatur menggunakan sistem kontrol dengan dua kondisi berupa mode kontrol *On-Off*, sehingga respon sistem pada proses di *plant* mengalami penurunan *level* permukaan air yang relatif cepat.

Dari keseluruhan *running* sistem secara umum, sistem pengendalian ini telah beroperasi normal pada tiap-tiap instrumen dan elemen kontrolnya. Kendati hanya terdapat kendala secara proses yang harus menyesuaikan dengan sistem kontrol yang ada.



*Halaman ini sengaja dikosongkan*

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil dari perancangan dan pembuatan sistem pengendalian *level* pada tangki pemanas *plant fluidized bed reactor*, maka dapat diambil kesimpulan sebagai berikut :

1. Telah dibuat rancang bangun sistem pengendalian *level* pada tangki pemanas, dengan adanya sistem tersebut maka fluida yang masuk pada tangki pemanas dapat dijaga ketinggiannya.
2. Telah dilakukan pengambilan data *open loop*, *close loop* dan uji *disturbance*. Pada *open loop*, dapat dilihat bahwa proses tersebut memerlukan *controler* untuk menjamin agar proses dapat berjalan dengan baik. Pada *closed loop*, dapat diketahui bahwa sistem dari tangki pemanas membutuhkan pengendalian agar dapat menjaga *set point* sebesar 10 cm dengan *rise time* sebesar 22,52 detik dan pada uji *disturbance* diketahui bahwa performansi pengendali tersebut dalam mempertahankan *set point* terhadap perubahan sangat baik.

#### **5.2 Saran**

Dari hasil perancangan dan pembuatan sistem ada beberapa hal yang perlu diperhatikan, yaitu:

1. Perlu adanya separator agar proses pemisahan antara minyak dengan air dapat menghasilkan produk yang lebih baik.
2. Perlu adanya perawatan dan pemeliharaan agar alat dapat bekerja dengan baik.

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- [1] Arindya, Radita. 2014. *Instrumentasi dan Kontrol Proses*. Yogyakarta : Graha Ilmu.
- [2] Zulmi, Faizal. *Rancang Bangun Alat Pendeteksi Jarak Pada Kendaraan Berbasis Arduino*. Teknik Elektro Universitas Mercu Buana. Jakarta.
- [3] Nanang Budi Hartono, Kemalasari, dkk. 2014. *Pengaturan Posisi Motor Servo Dc dengan Metode P, PI, PID*. Surabaya : Politeknik Elektronika Negeri Surabaya
- [8] Anonymous. 2014. *Pengertian Boiler Ketel uap* [diakses dari : <http://blog.unnes.ac.id/antosupri/pengertian-boiler-ketel-uap/>]
- [5] Lely Kojayanti, Martinus, dkk. 2014. *Reaksi Jamak pada Fluidized Bed Membran Reactor*. Jakarta : Fakultas Teknik Universitas Indonesia
- [6] Jazz, Rev. 2014. *Jenis-jenis Pompa Berdasarkan Cara Kerjanya Mengalirkan Fluida*. [diakses dari : [www.prosesindustri.com/2014/12/jenis-jenis-pompa-berdasarkan-cara-kerjanya-mengalirkan-fluida.html](http://www.prosesindustri.com/2014/12/jenis-jenis-pompa-berdasarkan-cara-kerjanya-mengalirkan-fluida.html)]
- [7] Anonymous. 2012s. *Pengertian, cara kerja dan jenis-jenis Motor Servo*. [diakses dari : [www.partner3d.com/motor-servo-pengertian-cara-kerja-dan-jenis-jenisnya/](http://www.partner3d.com/motor-servo-pengertian-cara-kerja-dan-jenis-jenisnya/)]
- [8] Winoto, Ardi. 2009. *Mikrokontroller AVR dan Pemrogramannya dengan Bahasa C pada Win AVR*. Bandung : Teknik Informatika
- [9] Anonymous. 2012. *LCD (Liquid Cristal Display)*. [diakses dari : <http://elektronika-dasar.web.id/lcd-liquid-cristal-display/>]
- [10] Satriya Dwi Ariffudin. 2014. *Perancangan Sistem Pemanas pada Rancang Bangun Mesin Pengaduk Bahan Baku Sabun Mandi Cair*. Surabaya : Universitas Negeri Surabaya

## *Lampiran A.*

### *Datasheet Level Switch*

# Stem Mounted Multi-Point Level Switch FL Series



The FL Magnetic level switch is designed to meet demanding customer applications for liquid level sensing in rugged hostile environments. Each FL is manufactured to the users specifications making it fit to work precisely according to the application requirements.

#### Features

- Custom tailored to user specifications
- Long switch life
- Up to seven switch points

# FL Series Specifications

## Performance

- FLR: 7 Switch Points  
Min. distance between levels: 1.75 in. (45mm)
- FLE: 3 Switch Points

## Environmental

- Operating Temperature:  
14° to 212°F (-10° to 100°C)

## Certification

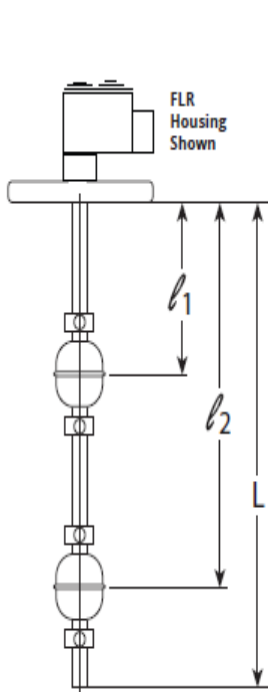
- FLE:  
NEMA 4 & 7, IP65  
IP43
- FLR:  
NEMA 4 & 7, IP65

## Electrical

- Switch Rating:  
FLE: 20 VA, 50 VA  
FLR: 50 VA, 180 VA
- Max Current:  
0.5 A AC
- Max Voltage:  
220 VAC

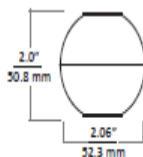
## Physical

- Stem Length:  
FLE: 24 in. (610 mm)  
FLR: 153 in. (3900 mm)
- Cable Entry:  
FLE: 3/4 in.  
FLR: 3/4 in.

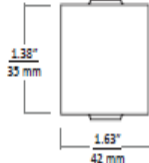


### FLR Floats

#### Float AA

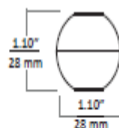


#### Float BB

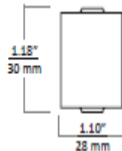


### FLE Floats

#### Float SS



#### Float KK



# FLR Model Configuration Options

Model Number: FLR- \_\_\_\_\_  
                                  A    B    C    D    E    F    G    H    I

## A. Mounting Type

- 0** Flat face flange
- 3** Triclamp
- 4** Plug mounted from outside of tank

## B. Mounting Option and Size

- A\_\*** Ansi flange 150# (size=2, 2.5, 3)
- SF\_\*** Triclamp (size=2)
- T\_\*** NPT plug (size=1.5, 2, 2.5, 3)

\*Note: Add an 'S' after mount size for Slide Connection

## C. Material

- S6** 316L SS

## D. Housing

- H3** NEMA 4 & 7, IP65 (3/4 in. cable entry)
- W\_** No housing, mounting option with 3/4 NPT plug and 12 or 36 in. lead wires

## E. Reed Switch

- A** 50 VA
- C** 180 VA

## F. Number of Switch Points

- 1-7** Select the number of switch points required

## G. Number of Floats

- 1-7** Select the number of floats required

## H. Float Type

- AA** 316L SS (2.06 in. diameter, 0.59 SG)
- AA.92** 316L SS (2.06 in. diameter, 0.92 SG)
- BB** 316L SS (1.63 in. diameter, 0.607 SG)
- BB.92** 316L SS (1.63 in. diameter, 0.92 SG)

## I. Probe Length (in./mm)

- L\_** in./mm (up to 153 in / 3900 mm)

## Switch Point Location(s)

(Measured from process connection)

- 1** \_\_\_\_ in./mm (designate NO or NC position)
- 2** \_\_\_\_ in./mm (designate NO or NC position)
- 3** \_\_\_\_ in./mm (designate NO or NC position)
- 4** \_\_\_\_ in./mm (designate NO or NC position)
- 5** \_\_\_\_ in./mm (designate NO or NC position)
- 6** \_\_\_\_ in./mm (designate NO or NC position)
- 7** \_\_\_\_ in./mm (designate NO or NC position)

## Lampiran B.

### Datasheet Mikrokontroler Atmega 32

#### Features

- High-performance, Low-power Atmel®AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 × 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 32Kbytes of In-System Self-programmable Flash program memory
  - 1024Bytes EEPROM
  - 2Kbytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
  - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    - 8 Single-ended Channels
    - 7 Differential Channels in TQFP Package Only
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad QFNMLF
- Operating Voltages
  - 2.7V - 5.5V for ATmega32L
  - 4.5V - 5.5V for ATmega32
- Speed Grades
  - 0 - 8MHz for ATmega32L
  - 0 - 16MHz for ATmega32
- Power Consumption at 1MHz, 3V, 25°C
  - Active: 1.1mA
  - Idle Mode: 0.35mA
  - Power-down Mode: < 1µA



8-bit **AVR**<sup>®</sup>  
Microcontroller  
with 32KBytes  
In-System  
Programmable  
Flash

ATmega32  
ATmega32L

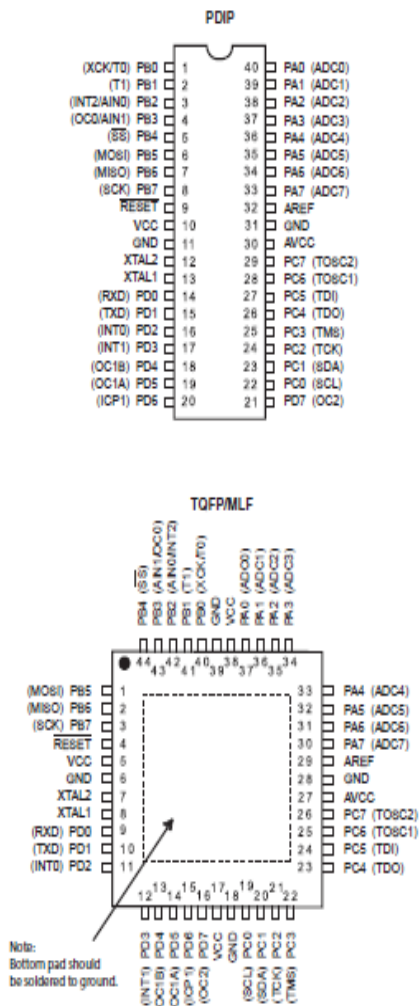
2503Q-AVR-02/11





## Pin Configurations

Figure 1. Pinout ATmega32



The Atmel®AVR®AVR core combines a rich Instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega32 provides the following features: 32Kbytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 1024bytes EEPROM, 2Kbyte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire Interface, A/D Converter, SRAM, Timer/Counters, SPI port, and Interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega32 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The Atmel AVR ATmega32 is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

## Pin Descriptions

**VCC** Digital supply voltage.

**GND** Ground.

**Port A (PA7..PA0)** Port A serves as the analog inputs to the A/D Converter.

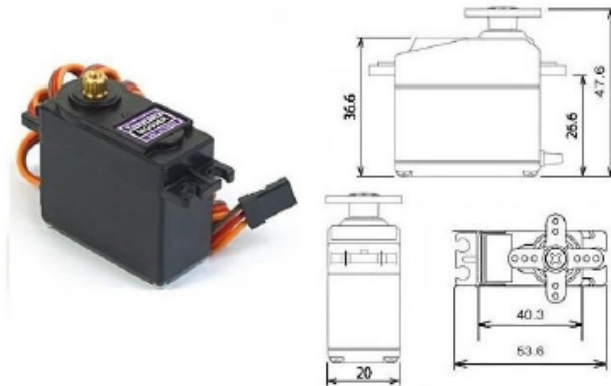
Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

<b>Port B (PB7..PB0)</b>	<p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port B also serves the functions of various special features of the ATmega32 as listed on <a href="#">page 57</a>.</p>
<b>Port C (PC7..PC0)</b>	<p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.</p> <p>The TD0 pin is tri-stated unless TAP states that shift out data are entered.</p> <p>Port C also serves the functions of the JTAG interface and other special features of the ATmega32 as listed on <a href="#">page 60</a>.</p>
<b>Port D (PD7..PD0)</b>	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega32 as listed on <a href="#">page 62</a>.</p>
<b>RESET</b>	Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in <a href="#">Table 15 on page 37</a> . Shorter pulses are not guaranteed to generate a reset.
<b>XTAL1</b>	Input to the Inverting Oscillator amplifier and input to the Internal clock operating circuit.
<b>XTAL2</b>	Output from the Inverting Oscillator amplifier.
<b>AVCC</b>	AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to $V_{CC}$ , even if the ADC is not used. If the ADC is used, it should be connected to $V_{CC}$ through a low-pass filter.
<b>AREF</b>	AREF is the analog reference pin for the A/D Converter.

## Lampiran C.

### Datasheet Motor Servo MG 996R Tower Pro

## MG996R High Torque Metal Gear Dual Ball Bearing Servo



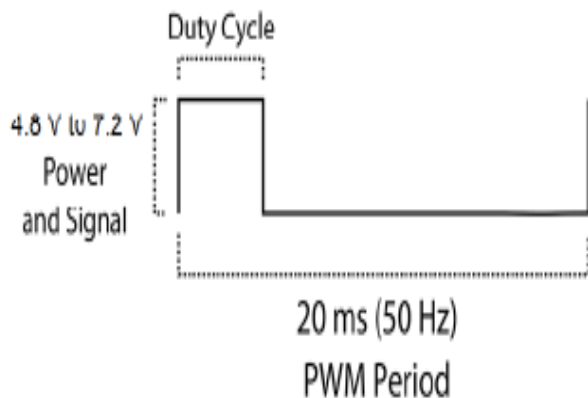
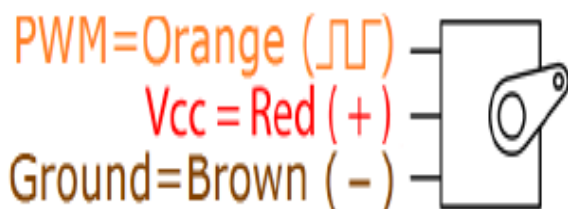
This High-Torque MG996R Digital Servo features metal gearing resulting in extra high 10kg stalling torque in a tiny package. The MG996R is essentially an upgraded version of the famous MG995 servo, and features upgraded shock-proofing and a redesigned PCB and IC control system that make it much more accurate than its predecessor. The gearing and motor have also been upgraded to improve dead bandwidth and centering. The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

This high-torque standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG996R Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

#### Specifications

- Weight 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 9.4 kgf-cm (4.8 V), 11 kgf-cm (6 V)
- Operating speed: 0.17 s/60° (4.8 V), 0.14 s/60° (6 V)

- Operating voltage: 4.8 V a 7.2 V
- Running Current 500 mA - 900 mA (6V)
- Stall Current 2.5 A (6V)
- Dead band width: 5  $\mu$ s
- Stable and shock proof double ball bearing design
- Temperature range: 0  $^{\circ}$ C - 55  $^{\circ}$ C



## ***Lampiran D.***

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

```
This program was created by the  
CodeWizardAVR V3.12 Advanced  
Automatic Program Generator  
© Copyright 1998-2014 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com
```

```
Project : Oil Sludge  
Version :  
Date   : 23/05/2017  
Author :  
Company :  
Comments:
```

```
Chip type       : ATmega32  
Program type    : Application  
AVR Core Clock frequency: 8.000000 MHz  
Memory model    : Small  
External RAM size : 0  
Data Stack size : 512
```

```
*****
```

```
**/
```

```
#include <mega32.h>  
#include <delay.h>  
#include <stdlib.h>
```

```
// Alphanumeric LCD functions  
#include <alcd.h>
```

```
#define thermocouple1 PORTA.0  
#define thermocouple2 PORTA.1
```

```

#define trigger_ultra PORTA.7
#define echo_ultra  PINA.6

#define ButtonUp    PINA.2
#define ButtonDown  PINA.3
#define ButtonRight PINA.4
#define ButtonOK    PINA.5

#include "serial.c"
// Declare your global variables here
unsigned char x, servo_ultra, servo_ther1, servo_ther2;
float suhu1, suhu2;
int flag_set,set=0, state;
float suhu; //ketinggian1,ketinggian=0.0;
int  setpoint_ketinggian= 10,  setpoint_suhu1=100,
setpoint_suhu2=70; //setpoint default
int setpoint_suhu11,setpoint_suhu22,setpoint_ketinggiann;
unsigned char  lcd_buffer1[16],  lcd_buffer2[16],
lcd_buffer3[16],lcd_buffer4[16];
unsigned char cetak_s1[16],cetak_s2[16],cetak_s3[16];
enum menu{t1,s1,s2};
#include <stdio.h>

//kontrol servo
// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Reinitialize Timer 0 value
TCNT0=0xB2;
// Place your code here
  x++;

  if(x>=servo_ultra)PORTB.0 = 0;
  else PORTB.0 = 1;
}

```

```

if(x>=servo_ther1)PORTB.1 = 0;
else PORTB.1 = 1;

if(x>=servo_ther2)PORTB.2 = 0;
else PORTB.2 = 1;

if(x>=255){x=0;}
}

// SPI functions
#include <spi.h>

void read_thermocouple(){
    unsigned int baca_suhu;
    baca_suhu = (unsigned) spi(0)<<8; //baca byte MSB
    //baca LSB dengan SPI dan gabungkan dengan MSB
    baca_suhu |= spi(0);
    thermocouple1 = 1; //baca sensor dimatikan
    thermocouple2 = 1;
    delay_ms(200);
    if(ButtonOK==0){flag_set++;                delay_ms(1000);
set=flag_set%2;}
    //hitung tegangan dalam mV
    suhu = (((float)baca_suhu*5000)/4096)/40);

}
//
//void HCSR04Trigger()
//{
//    trigger_ultra=1;    //high
//    delay_us(10);    //wait 10uS
//    trigger_ultra=0;    //low
//}
//
//void read_ultrasonic()
//{

```



```

// long i,result;
//
// //Send a trigger pulse
// HCSR04Trigger();
//
// //menunggu hingga echo rising egde 25uS
// for(i=0;i<600000;i++){
//     if(echo_ultra!=1)
//         continue; //Line is still low, so wait
//     else
//         break; //High edge detected, so break.
// }
//
// if(i==600000)
//     ketinggian=1000000; //Indicates time out
//
// //High Edge Found
//
// //Setup Timer1
// TCCR1A=0X00;
// TCCR1B=0x01;
// TCNT1=0x00; //Init counter
//
// //Now wait for the falling edge
// for(i=0;i<600000;i++){
//     if(echo_ultra==1)
//     {
//         if(TCNT1> 60000) break; else continue;
//     }
//     else
//         break;
// }
//
// if(i==600000) // 60ms
//     ketinggian=1000000; //Indicates time out
//

```

```

// //Falling edge found
//
// result=TCNT1;
//
// //Stop Timer
// TCCR1B=0x00;
//
// if(result>60000) // 60ms dan maksimal 38ms
//   ketinggian=10000000; //No obstacle
// else{
//   ketinggian=((result)*(0.034/2)/8)-20)*-1;
//   } // /58/8 = *(0.034/2)/8
//}

//pengaturan tampil 3 digit value
void digit_3(int a, int xx, int yy){
if(a >= 100){lcd_gotoxy(xx,yy);}
if(a < 100 && a >= 10){lcd_gotoxy(xx,yy); lcd_puts("
");lcd_gotoxy(xx+1,yy);}
if(a < 10) {lcd_gotoxy(xx,yy); lcd_puts("
");lcd_gotoxy(xx+2,yy);}
}

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Function: Bit7=Out Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In
Bit1=Out Bit0=Out
DDRA=(1<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) |
(0<<DDA3) | (0<<DDA2) | (1<<DDA1) | (1<<DDA0);
// State: Bit7=0 Bit6=0 Bit5=P Bit4=P Bit3=P Bit2=P Bit1=0
Bit0=0

```

```
PORTA=(0<<PORTA7) | (0<<PORTA6) | (1<<PORTA5) |  
(1<<PORTA4) | (1<<PORTA3) | (1<<PORTA2) | (0<<PORTA1)  
| (0<<PORTA0);
```

```
// Port B initialization
```

```
// Function: Bit7=Out Bit6=In Bit5=Out Bit4=Out Bit3=In  
Bit2=Out Bit1=Out Bit0=Out
```

```
DDRB=(1<<DDB7) | (0<<DDB6) | (1<<DDB5) | (1<<DDB4) |  
(0<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
```

```
// State: Bit7=0 Bit6=T Bit5=0 Bit4=0 Bit3=T Bit2=0 Bit1=0  
Bit0=0
```

```
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) |  
(0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1)  
| (0<<PORTB0);
```

```
// Port C initialization
```

```
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In  
Bit1=In Bit0=In
```

```
DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) |  
(0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
```

```
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T  
Bit0=T
```

```
PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) |  
(0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1)  
| (0<<PORTC0);
```

```
// Port D initialization
```

```
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In  
Bit1=In Bit0=In
```

```
DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) |  
(0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);
```

```
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T  
Bit0=T
```

```
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) |  
(0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1)  
| (0<<PORTD0);
```

```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 1000.000 kHz
// Mode: Normal top=0xFF
// OC0 output: Disconnected
// Timer Period: 0.078 ms
TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) |
(0<<WGM01) | (0<<CS02) | (1<<CS01) | (0<<CS00);
TCNT0=0xB2;
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Disconnected
// OC1B output: Disconnected
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) |
(0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) |
(0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
```

```

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0<<AS2;
TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) |
(0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) |
(0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) |
(1<<TOIE0);

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
MCUCSR=(0<<ISC2);

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) |
(0<<DOR) | (0<<UPE) | (0<<U2X) | (0<<MPCM);
UCSRB=(1<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) |
(1<<RXEN) | (1<<TXEN) | (0<<UCSZ2) | (0<<RXB8) |
(0<<TXB8);

```

```
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) |  
(0<<UPM0) | (0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) |  
(0<<UCPOL);  
UBRRH=0x00;  
UBRRL=0x33;
```

```
// Analog Comparator initialization  
// Analog Comparator: Off  
// The Analog Comparator's positive input is  
// connected to the AIN0 pin  
// The Analog Comparator's negative input is  
// connected to the AIN1 pin  
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) |  
(0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);  
SFIOR=(0<<ACME);
```

```
// ADC initialization  
// ADC disabled  
ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) |  
(0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) |  
(0<<ADPS0);
```

```
// SPI initialization  
// SPI Type: Master  
// SPI Clock Rate: 2764.800 kHz  
// SPI Clock Phase: Cycle Start  
// SPI Clock Polarity: Low  
// SPI Data Order: MSB First  
SPCR=(0<<SPIE) | (1<<SPE) | (0<<DORD) | (1<<MSTR) |  
(0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);  
SPSR=(0<<SPI2X);
```

```
// TWI initialization  
// TWI disabled  
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) |  
(0<<TWEN) | (0<<TWIE);
```

```

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD
menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 20
lcd_init(20);
lcd_gotoxy(0,0);
lcd_puts("----TUGAS AKHIR----");
lcd_gotoxy(0,1);
lcd_puts("|  PENGOLAHAN  |");
lcd_gotoxy(0,2);
lcd_puts("|  OIL SLUDGE  |");
lcd_gotoxy(0,3);
lcd_puts("-----");

delay_ms(3000);
state=t1;
lcd_clear();
// Global enable interrupts
#asm("sei")

while (1)
{
if(start==1){ //connect PC
//if(start==0){ //untuk monitoring tanpa PC.
// Place your code here
if(ButtonOK==0){flag_set++; delay_ms(1000);
set=flag_set%2;} //button OK ditekan

```

```

lcd_gotoxy(1,1);
lcd_puts("LEVEL =");
lcd_gotoxy(1,2);
lcd_puts("SUHU R =");
lcd_gotoxy(1,3);
lcd_puts("SUHU C =");
lcd_gotoxy(17,1);
lcd_puts("cm");
lcd_gotoxy(17,2);
sprintf(lcd_buffer1,"%cC",0xDF);
lcd_puts(lcd_buffer1);
lcd_gotoxy(17,3);
sprintf(lcd_buffer1,"%cC",0xDF);
lcd_puts(lcd_buffer1);
lcd_gotoxy(16,3);
lcd_puts(" ");

//=====kon
figurasi setpoint=====
if(set==1){
    lcd_gotoxy(0,0);
    lcd_puts("-#- SETPOINT #-");
    digit_3(setpoint_ketinggian, 10, 1);
    ftoa(setpoint_ketinggian,2,lcd_buffer4);
    lcd_puts(lcd_buffer4);
    digit_3(setpoint_suhu1, 10, 2);
    ftoa(setpoint_suhu1,2,lcd_buffer2);
    lcd_puts(lcd_buffer2);
    digit_3(setpoint_suhu2, 10,3);
    ftoa(setpoint_suhu2,2,lcd_buffer3);
    lcd_puts(lcd_buffer3);
    switch(state)
    {
        case t1:
            lcd_gotoxy(0,1);lcd_puts(">");
            lcd_gotoxy(0,2);lcd_puts(" ");

```



```

        lcd_gotoxy(0,3);lcd_puts(" ");

        if(ButtonUp!=1                                &&
setpoint_ketinggian<10){setpoint_ketinggian++;delay_ms(300);}
        if(ButtonDown!=1                              &&
setpoint_ketinggian>5){setpoint_ketinggian--;delay_ms(300);}
        if(ButtonRight!=1){state=s1;delay_ms(1000);}
        break;

    case s1:
        lcd_gotoxy(0,1);lcd_puts(" ");
        lcd_gotoxy(0,2);lcd_puts(">");
        lcd_gotoxy(0,3);lcd_puts(" ");

        if(ButtonUp!=1                                &&
setpoint_suhu1<200){setpoint_suhu1++;delay_ms(300);}
//maks setpoint r =250 (ubah angka)
        if(ButtonDown!=1                              &&
setpoint_suhu1>20){setpoint_suhu1--;delay_ms(300);} //maks
setpoint r =100 (ubah angka)
        if(ButtonRight!=1){state=s2;delay_ms(1000);}
        break;

    case s2:
        lcd_gotoxy(0,1);lcd_puts(" ");
        lcd_gotoxy(0,2);lcd_puts(" ");
        lcd_gotoxy(0,3);lcd_puts(">");

        if(ButtonUp!=1                                &&
setpoint_suhu2<100){setpoint_suhu2++;delay_ms(300);}
        if(ButtonDown!=1                              &&
setpoint_suhu2>20){setpoint_suhu2--;delay_ms(300);}
        if(ButtonRight!=1){state=t1;delay_ms(1000);}
        break;
    }

```

```

}
//=====monitoring suhu dan
kontrol=====
if(set==0){
    setpoint_suhu11= setpoint_suhu1*50/100;
    setpoint_suhu22= setpoint_suhu2*5/100;
    setpoint_ketinggiann= setpoint_ketinggian*50/100;

    lcd_gotoxy(0,1); lcd_puts(" ");
    lcd_gotoxy(0,2); lcd_puts(" ");
    lcd_gotoxy(0,3); lcd_puts(" ");
    lcd_gotoxy(0,0);
    lcd_puts("-#- ACTUAL -#-");

    //baca sensor suhu1 dan tampil lcd
    thermocouple1 = 0;
    thermocouple2 = 1;
    read_thermocouple();
    suhu1=suhu;
    digit_3(suhu1, 10, 2);
    ftoa(suhu1,2,lcd_buffer2);
    lcd_puts(lcd_buffer2);

    //baca sensor suhu2 dan tampil lcd
    thermocouple1 = 1;
    thermocouple2 = 0;
    read_thermocouple();
    suhu2=suhu+0,7; //((suhu+1.8)*0.9593+1.0159; //kalibrasi
suhu2
    digit_3(suhu2, 10, 3);
    ftoa(suhu2,2,lcd_buffer3);
    lcd_puts(lcd_buffer3);

    //baca sensor ketinggian dan tampil lcd
    // read_ultrasonic();
    // ketinggian1=ketinggian; //kalibrasi ketinggian

```

```

//    if(ketinggian1>1000){ }
//    else{
//        if(ketinggian1<=0) ketinggian1=0;
//        digit_3(ketinggian1, 10, 1);
//        ftoa(ketinggian1,2,lcd_buffer4);
//        lcd_puts(lcd_buffer4);
//    }
//    //kontrol servo ketinggian
//    if(ketinggian1 < setpoint_ketinggian){    //kurang dari
setpoint
        if(PINA.6 == 0){
            //buka servo
            servo_ultra = 9;
            //delay_ms(100);
            //lcd_clear();
            lcd_gotoxy(11,1);
            lcd_puts("00.00");
        }

//    else if(ketinggian1 > setpoint_ketinggiann && ketinggian1
//    <= setpoint_ketinggian)
//        { //ketinggian berada diantara setpoint dan 50% dari
setpoint
//            servo_ultra = 19 + (12/((setpoint_ketinggian-
setpoint_ketinggiann)/(ketinggian1-setpoint_ketinggiann));
//        }

        else{ //tutup servo
            servo_ultra = 31;
//            lcd_gotoxy(11,1);
//            digit_3(setpoint_ketinggian, 10, 1);
//            ftoa(setpoint_ketinggian,2,lcd_buffer4);
//            lcd_puts("10.00");
//            //delay_ms(10);

```

```

        //lcd_clear();
        lcd_gotoxy(11,1);
//      digit_3(setpoint_ketinggian, 10, 1);
//      ftoa(setpoint_ketinggian,2,lcd_buffer4);
        lcd_puts("10.00");
    }

    //kontrol servo suhu1
    if(suhu1 <= setpoint_suhu1){
        //tutup servo
        servo_ther1 = 25;
    }
//      else if(suhu1 > setpoint_suhu11 && suhu1 <=
setpoint_suhu1)
//      { //suhu berada diantara setpoint dan 50% dari setpoint
//          servo_ther1 = 24 - (15/((setpoint_suhu1-
setpoint_suhu11)/(suhu1-setpoint_suhu11)));
//      }
    else
    { //buka servo
        servo_ther1 = 7;
    }
    //kontrol servo suhu2
    if(suhu2 < (setpoint_suhu2)){ //-setpoint_suhu22){
        //tutup servo
        servo_ther2 = 31,5;
    }

    // else if(suhu2 >= (setpoint_suhu2-setpoint_suhu22) &&
suhu2 < (setpoint_suhu2+setpoint_suhu22)){
        //1/2 servo
        // servo_ther2 = 26;
    // }
//      else if(suhu2 > setpoint_suhu22 && suhu2 <=
setpoint_suhu2)
//      { //suhu berada diantara setpoint dan 50% dari setpoint

```

```

//          servo_ther2 = 26 - (7/((setpoint_suhu2-
setpoint_suhu22)/(suhu2-setpoint_suhu22)));
//      }
else
{
    //buka servo
    servo_ther2 = 19;//
}
//=====tampil ke PC=====
printf("%s",lcd_buffer2);
putchar(',');
printf("%s",lcd_buffer3);
putchar(',');
ftoa(setpoint_ketinggian,2,cetak_s3);
//if(PINA.6==1) printf("%s",cetak_s3);
//else puts("0.00");
putchar(',');
ftoa(setpoint_suhu1,2,cetak_s1);
printf("%s",cetak_s1);
putchar(',');
ftoa(setpoint_suhu2,2,cetak_s2);
printf("%s",cetak_s2);
putchar(',');
printf("%s",cetak_s3);
putchar(13);
putchar(10);
//=====
}
}
else{lcd_clear(); lcd_gotoxy(0,0); lcd_putsf("Connect PC");}
}
}

```

## BIODATA PENULIS



Penulis dilahirkan di Kota Surabaya 16 Agustus 1996. Alamat asli rumah di Sidoarjo Perumahan Griya Permata Hijau, N-18. Pada tahun 2008, penulis menyelesaikan pendidikan di SDN Balongsari 1 Mojokerto. Pada tahun 2011 menyelesaikan pendidikan tingkat menengah pertama di SMPN 4 Mojokerto. Tahun 2014 menyelesaikan pendidikan di SMAN 2 Mojokerto. Pada tahun 2017 ini, penulis mampu menyelesaikan gelar ahli madya di Program Studi D3 Teknik Instrumentasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya. Penulis berhasil menyelesaikan Tugas Akhir dengan judul “Rancang Bangun Sistem Pengendalian *Level* Pada Tangki Pemanas *Plant Fluidized Bed Reactor* Pengolahan *Oil Sludge*”. Bagi pembaca yang memiliki kritik dan saran mengenai tugas akhir ini, dapat menghubungi penulis melalui email [putrimerlisa26@gmail.com](mailto:putrimerlisa26@gmail.com).