



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE 145561**

**RANCANG BANGUN SISTEM *MONITORING VOLUME* LUMPUR  
PADA *MUDTANK* BERBASIS MIKROKONTROLER**

Yudha Bintara Khresna Teguh Ary Arbajaya  
NRP 1031150000048

Pembimbing

Ir. Djoko Suprajitno Rahardjo, M.T.  
Ciptian Weried Priananda, S.ST., M.T.  
Sopar Simanullang, S.T.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

***FINAL PROJECT - TE 145561***

***DESIGN OF MONITORING SYSTEM MUD VOLUME IN  
MUDTANK BASED ON MICROCONTROLLER***

Yudha Bintara Khresna Teguh Ary Arbajaya  
NRP 1031150000048

*Supervisor*

Ir. Djoko Suprajitno Rahardjo, M.T.  
Ciptian Weried Priananda, S.ST., M.T.  
Sopar Simanullang, S.T.

DEPARTMENT OF ELECTRICAL AUTOMATION ENGINEERING  
Faculty of Vocations  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018



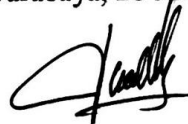
## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Rancang Bangun Sistem *Monitoring* Volume Lumpur pada Mudtank Berbasis Mikrokontroler**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 26 Juli 2018



**Yudha Bintara K.T.A.A**  
NRP 10311500000048

\

-----Halaman ini sengaja dikosongkan-----

# RANCANG BANGUN SISTEM MONITORING VOLUME LUMPUR PADA MUDTANK BERBASIS MIKROKONTROLER

## TUGAS AKHIR

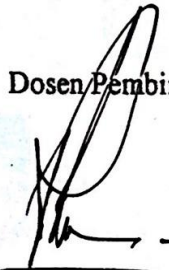
Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya Teknik  
Pada


Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II

  
Ir. Dioko Suprajitno R., M.T.  
NIP. 195506221987011001

  
Ciptan Weried P., S.ST., M.T.  
NIP. 1990201711060

SURABAYA  
JULI, 2018

-----Halaman ini sengaja dikosongkan-----



# **RANCANG BANGUN SISTEM *MONITORING* VOLUME LUMPUR PADA MUDTANK BERBASIS MIKROKONTROLER**

**Nama** : Yudha Bintara Khresna Teguh Ary Arbajaya  
**Pembimbing I** : Ir. Djoko Suprajitno Rahardjo, M.T.  
**Pembimbing II** : Ciptian Weried Priandana, S.ST., M.T.  
**Pembimbing III** : Sopar Simanullang, S.T.

## **ABSTRAK**

Pengeboran merupakan suatu kegiatan melubangi permukaan tanah untuk mendapatkan sumber air, gas atau minyak. Untuk menunjang proses pengeboran, terdapat beberapa sistem yaitu sistem tenaga, sistem pengangkat, sistem pemutar, sistem sirkulasi lumpur dan sistem pencegahan semburan. Sistem sirkulasi lumpur merupakan proses pengaliran lumpur pengeboran dari peralatan sirkulasi lumpur menuju rangkaian pipa bor. Lumpur pengeboran dapat berfungsi sebagai penahan fluida formasi (tekanan formasi). Saat proses pengeboran, lumpur yang dihasilkan harus sesuai dengan volume lubang pengeboran agar tidak terjadi kelebihan atau kekurangan lumpur pada saat proses berlangsung, sehingga tekanan formasi yang terdapat dalam bumi dapat ditahan.

Pada penelitian ini membuat suatu alat yang dapat mengukur volume lumpur pada *mud tank* agar diperoleh volume lumpur sesuai dengan yang dibutuhkan. Alat ini terdiri dari sensor pelampung yang diintegrasikan dengan mikrokontroler ATmega16, serta *Human Machine Interface* (HMI) sebagai *monitoring* keadaan volume lumpur.

Setelah melakukan pengujian, sistem *monitoring* volume lumpur telah mampu mengukur ketinggian dan volume lumpur dengan tingkat keakuratan sebesar  $\pm 99,15$  %. Dari hasil tersebut, kesalahan pengukuran volume lumpur pada *mudtank* yang digunakan saat proses pengeboran berlangsung dapat terminimalisasi.

**Kata Kunci** : *monitoring*, pengeboran, sirkulasi lumpur, volume lumpur.

-----Halaman ini sengaja dikosongkan-----

# ***DESIGN OF MONITORING SYSTEM VOLUME MUD IN MUDTANK BASED ON MICROCONTROLLER***

***Name*** : Yudha Bintara Khresna Teguh Ary Arbajaya  
***Supervisor I*** : Ir. Djoko Suprajitno Rahardjo, M.T.  
***Supervisor II*** : Ciptian Weried Priananda, S.ST., M.T.  
***Supervisor III*** : Sopar Simanullang, S.T.

## ***ABSTRACT***

*Drilling is a hole in the ground to get a source of water, gas or oil. To support the drilling process, there are several systems: power system, lifting system, rotating system, mud circulation system and blow prevention system. The mud circulation system is a drilling mud drilling process from the mud flow equipment to the drill pipe line. Drilling mud can serve as a formation fluid holder (formation pressure). During the drilling process, the resulting sludge must be in accordance with the volume of the drill holes to avoid excess or shortage of sludge during the process, so that formation pressures within the earth can be retained.*

*In this study, a tool that can measure the volume of mud in mud tanks to obtain the volume of mud in accordance with the required. This tool consists of a buoy sensor that is integrated with ATmega16 microcontroller, and Human Machine Interface as monitoring the state of mud volumes.*

*After conducting the test, the mud volume monitoring system has been able to measure the height and volume of the mud with an accuracy level of  $\pm 99,15\%$ . From these results, the measurement error of the sludge volume in mudtank used during the drilling process can be minimized.*

***Keywords*** : monitoring, drilling, mud circulation, mud volumes.

-----Halaman ini sengaja dikosongkan-----

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

### **RANCANG BANGUN SISTEM *MONITORING* VOLUME LUMPUR PADA MUDTANK BERBASIS MIKROKONTROLER**

Dalam Tugas Akhir ini, dirancang alat *monitoring* volume lumpur yang digunakan pada saat pengeboran berlangsung.

Penyusun mengucapkan terimakasih utamanya kepada kedua orang tua yang memberi dukungan moril dan doa beliau berdua sangat berarti, serta semua pihak yang telah mendukung dalam pembuatan buku tugas akhir ini, termasuk dosen pembimbing kami, yaitu Bapak Ir. Djoko Suprajitno Rahardjo, M.T., Bapak Ciptian Weried Priananda, S.ST., M.T dan Bapak Sopar Simanullang, S.T yang telah menyempatkan waktunya untuk membimbing kami dalam menyelesaikan tugas akhir ini. Tidak lupa penulis ucapkan pada Alm. Bapak, Ibu dan Kakak saya, Keluarga Besar Nias, Keluarga Besar Girilaya, Adinda Berliana Putri serta teman-teman angkatan 2015 atas bantuan dan dukungan semangatnya serta semua pihak yang telah membantu dan tidak dapat penulis sebutkan satu persatu.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 26 Juli 2018

Penulis

-----Halaman ini sengaja dikosongkan-----

## DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR	<b>Error! Bookmark not defined.</b>
HALAMAN PENGESAHAN INSTITUT	<b>Error! Bookmark not defined.</b>
HALAMAN PENGESAHAN PERUSAHAAN	<b>Error! Bookmark not defined.</b>
ABSTRAK.....	ix
ABSTRACT.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL.....	xxiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Permasalahan.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Metodologi Penelitian.....	3
1.6 Sistematika Laporan.....	4
BAB II <i>MONITORING</i> VOLUME LUMPUR.....	7
2.1 Sistem Sirkulasi Lumpur.....	7
2.2 Pengukuran Ketinggian Permukaan dan Volume.....	8
2.3 Mikrokontroler ATmega16.....	10
2.3.1 Arsitektur ATMEGA16.....	11
2.3.2 Konfigurasi Pin ATMEGA16.....	11
2.4 Sensor Ketinggian.....	15
2.5 Pengondisian Sinyal <i>Analog</i> .....	17
2.5.1 Perubahan Level Sinyal.....	18
2.6 Rangkaian Jembatan <i>Wheatstone</i> dan Potensiometer.....	18
2.6.1 Rangkaian Jembatan <i>Wheatstone</i> .....	18

2.6.2 Rangkaian Potensiometer.....	21
2.7 IC LM741.....	23
2.8 Catu Daya atau <i>Power supply</i> .....	24
2.8.1 Prinsip Kerja DC <i>Power supply</i> .....	26
2.9 Komunikasi Serial RS485.....	29
2.10 <i>Visual Basic .NET</i> .....	30
2.11 <i>CodeVision AVR</i> .....	31
2.12 <i>Database My SQL</i> .....	32
BAB III PERANCANGAN <i>MONITORING VOLUME LUMPUR</i> .....	36
3.1 Konfigurasi Sistem.....	36
3.2 Perancangan Perangkat Keras Mekanik.....	38
3.2.1 Perancangan Mekanik Sensor <i>Level</i> .....	38
3.2.2 Perancangan Mekanik Sistem Minimum.....	40
3.2.3 Perancangan Mekanik Keseluruhan.....	41
3.3 Perancangan Perangkat Keras Elektrik.....	42
3.3.1 Perancangan <i>Power supply</i> .....	42
3.3.2 Perancangan Rangkaian Jembatan <i>Wheatstone</i> dan ASC43	
3.3.3 Perancangan Sistem Minimum Mikrokontroler ATMega16.....	47
3.3.4 Perancangan Komunikasi Serial RS485 ke USB.....	48
3.4 Perancangan Perangkat Lunak.....	49
3.4.1 Algoritma Pembacaan Sensor Ketinggian dan Komunikasi RS485.....	50
3.4.2 Program Pembacaan Sensor Ketinggian.....	51
3.4.3 Perancangan <i>Human Machine Interface (HMI)</i> .....	51
1.4.3.1 Perancangan <i>Visual Monitoring HMI</i> .....	51
1.4.3.2 Flowchart HMI.....	54
BAB IV HASIL PENGUJIAN SISTEM <i>MONITORING VOLUME</i> LUMPUR.....	60
4.1 Pengujian Perangkat Keras Elektrik.....	60
4.1.1 Pengujian Rangkaian <i>Power supply</i> .....	60
4.1.2 Pengujian Sensor Level Potensiometer.....	63
4.1.3 Pengujian Rangkaian Jembatan <i>Wheatstone</i> dan <i>Analog</i> <i>Signal Conditioning (ASC)</i> .....	65



4.1.4	Pengujian Sistem Minimum ATmega16.....	68
4.2	Pengujian Perangkat Lunak.....	70
4.2.1	Pengujian Komunikasi Serial RS485.....	70
4.2.2	Pengujian <i>Human Machine Interface</i> (HMI).....	72
4.2.2.1	Pengujian Tampilan <i>Login</i> .....	72
4.2.2.2	Pengujian Tampilan <i>Monitoring Channel</i> .....	74
4.2.2.3	Pengujian Pengatur Nama Rig.....	75
4.2.2.4	Pengujian Koneksi Serial.....	76
4.2.2.5	Pengujian Penentuan Luas Media yang dipakai.....	78
4.2.2.6	Pengujian Data Ketinggian dan Volume dari Mikrokontroler.....	79
4.2.2.7	Pengujian Tampilan Grafik.....	80
4.2.2.8	Pengujian Penyimpanan <i>Database</i> .....	82
4.3	Pengujian Keseluruhan Sistem.....	85
	BAB V PENUTUP.....	92
	DAFTAR PUSTAKA.....	94
	LAMPIRAN.....	96
	RIWAYAT HIDUP PENULIS.....	142

-----Halaman ini sengaja dikosongkan-----

## DAFTAR GAMBAR

	HALAMAN
<b>Gambar 2.1</b> Susunan Kaki Pin ATmega16.....	12
<b>Gambar 2.2</b> Sensor Potensiometer.....	17
<b>Gambar 2.3</b> Jembatan DC <i>Wheatstone</i> .....	19
<b>Gambar 2.4</b> Rangkaian Dasar Potensiometer.....	21
<b>Gambar 2.5</b> DIP 8 pin LM741.....	23
<b>Gambar 2.6</b> Blok Diagram DC <i>Power supply</i> .....	27
<b>Gambar 2.7</b> Transformator / Trafo Step Down.....	27
<b>Gambar 2.8</b> Rangkaian Penyearah Sederhana.....	28
<b>Gambar 2.9</b> Rangkaian Penyearah Gelombang Penuh.....	28
<b>Gambar 2.10</b> Rangkaian Penyearah DC <i>Power supply</i> .....	29
<b>Gambar 2.11</b> Rangkaian Dasar IC <i>Voltage Regulator</i> .....	29
<b>Gambar 2.12</b> Tampilan Awal Visual Basic.....	30
<b>Gambar 2.13</b> Tampilan Awal Code Vision AVR.....	31
<b>Gambar 2.14</b> Halaman Menu PhpMyAdmin.....	33
<b>Gambar 2.15</b> Form Menu <i>Database</i> .....	33
<b>Gambar 2.16</b> Form Menu <i>Privileges</i> .....	33
<b>Gambar 2.17</b> Form Menu <i>Add User</i> .....	34
<b>Gambar 2.18</b> Form Menu <i>Add User 2</i> .....	35
<b>Gambar 3.1</b> Bagan Alur Kerja dari Alat.....	36
<b>Gambar 3.2</b> Perancangan Kotak Sensor Tampak Dalam.....	39
<b>Gambar 3.3</b> Perancangan Kotak Sensor Tampak Luar.....	39
<b>Gambar 3.4</b> Perancangan Tiang Pendiri Sensor.....	40
<b>Gambar 3.5</b> Perancangan Mekanik Sistem Minimum.....	40
<b>Gambar 3.6</b> Perancangan Mekanik Keseluruhan.....	41
<b>Gambar 3.7</b> Skematik Regulator Tegangan 5v, 9v dan -9v.....	43
<b>Gambar 3.8</b> Rangkaian <i>Wheatstone Bridge</i> .....	44
<b>Gambar 3.9</b> Rangkaian <i>Wheatstone Bridge to pin ASC</i> .....	45
<b>Gambar 3.10</b> Rangkaian <i>Analog Signal Conditioning</i> .....	45
<b>Gambar 3.11</b> Rangkaian Op-Amp 1.....	46
<b>Gambar 3.12</b> Rangkaian Op-Amp 2.....	46
<b>Gambar 3.13</b> Rangkaian Sistem Minimum ATmega16.....	47

<b>Gambar 3.14</b> Modul Rangkaian RS485.....	49
<b>Gambar 3.15</b> Flowchart Mikrokontroler.....	50
<b>Gambar 3.16</b> Program Mapping Sensor.....	51
<b>Gambar 3.17</b> Tampilan Form Login.....	52
<b>Gambar 3.18</b> Tampilan Form <i>Menu Channel</i> .....	53
<b>Gambar 3.19</b> Tampilan Form Grafik.....	53
<b>Gambar 3.20</b> Tampilan Form <i>Database</i> .....	54
<b>Gambar 3.21</b> <i>Flowchart Form Login</i> .....	55
<b>Gambar 3.22</b> <i>Flowchart Form Channel</i> .....	56
<b>Gambar 3.23</b> <i>Flowchart Form Grafik</i> .....	57
<b>Gambar 3.24</b> <i>Flowchart Form Database</i> .....	58
<b>Gambar 3.25</b> <i>Flowchart Form Logout</i> .....	59
<b>Gambar 4.1</b> Pengukuran Sensor Level Potensiometer.....	63
<b>Gambar 4.2</b> Tampilan Login Monitoring.....	72
<b>Gambar 4.3</b> Tampilan <i>Login Salah</i> .....	73
<b>Gambar 4.4</b> Tampilan Login Benar.....	73
<b>Gambar 4.5</b> Tampilan Monitoring <i>Channel</i> .....	74
<b>Gambar 4.6</b> Tombol Pengatur Nama Rig.....	75
<b>Gambar 4.7</b> Tampilan Sebelum Mengatur Nama Rig.....	75
<b>Gambar 4.8</b> Tampilan Sesudah Mengatur Nama Rig.....	76
<b>Gambar 4.9</b> Pengatur Sambungan <i>Port Com</i> .....	77
<b>Gambar 4.10</b> Tampilan Setelah Tombol <i>Start</i> Ditekan.....	77
<b>Gambar 4.11</b> Tampilan Saat Pemutusan <i>Port Serial</i> .....	78
<b>Gambar 4.12</b> Pengaturan Luas, Hi-Top, dan Hi-Limit.....	78
<b>Gambar 4.13</b> Pengukuran Data Ketinggian 2.....	79
<b>Gambar 4.14</b> Pengukuran Data Ketinggian ke-1.....	79
<b>Gambar 4.15</b> Tombol Menu Grafik.....	80
<b>Gambar 4.16</b> Tampilan Grafik <i>Volume</i> .....	81
<b>Gambar 4.17</b> Tampilan Grafik <i>Level</i> .....	82
<b>Gambar 4.18</b> Tampilan Tombol Menu Datalog.....	82
<b>Gambar 4.19</b> Tampilan Penyimpanan <i>Database</i> .....	83
<b>Gambar 4.20</b> Penyimpanan ke Excel (.xls).....	84
<b>Gambar 4.21</b> Konfirmasi Penghapusan Data Sensor.....	84
<b>Gambar 4.22</b> Tampilan Saat Data Sensor Telah Dihapus.....	84

<b>Gambar 4.23</b>	Tampilan Sebelum Ditekan <i>Refresh</i> .....	85
<b>Gambar 4.24</b>	Tampilan Sesudah Ditekan <i>Refresh</i> .....	85
<b>Gambar 4.25</b>	Pengujian Sistem Keseluruhan 1.....	86
<b>Gambar 4.26</b>	Pengukuran Ketinggian Ke-2.....	88
<b>Gambar 4.27</b>	Pengukuran Ketinggian Ke-1.....	88
<b>Gambar 4.28</b>	Pengukuran Volume Ke-1.....	90
<b>Gambar 4.29</b>	Pengukuran Volume Ke-2.....	91

-----Halaman ini sengaja dikosongkan-----

## DAFTAR TABEL

	HALAMAN
<b>Tabel 2.1</b> Fungsi Khusus <i>Port A</i> Atmega 16.....	12
<b>Tabel 2.2</b> Fungsi Khusus <i>Port B</i> Atmega 16.....	13
<b>Tabel 2.3</b> Fungsi Khusus <i>Port C</i> Atmega 16.....	14
<b>Tabel 2.4</b> Fungsi Khusus <i>Port D</i> Atmega 16.....	15
<b>Tabel 3.1</b> Data Jembatan <i>Wheatstone</i> yang Diinginkan.....	43
<b>Tabel 3.2</b> Data Tegangan ASC yang Diinginkan.....	45
<b>Tabel 4.1</b> Pengujian Tegangan Regulator 7805 (5 volt).....	61
<b>Tabel 4.2</b> Pengujian Tegangan Regulator 7809 (9 volt).....	61
<b>Tabel 4.3</b> Pengujian Tegangan Regulator 7909 (-9 volt).....	61
<b>Tabel 4.4</b> Pengujian Sensor Level Potensiometer.....	64
<b>Tabel 4.5</b> Pengujian Rangkaian Jembatan <i>Wheatstone</i> .....	66
<b>Tabel 4.6</b> Pengujian Rangkaian ASC.....	66
<b>Tabel 4.7</b> Pengujian Sistem Minimum.....	68
<b>Tabel 4.8</b> Pengujian Komunikasi Serial.....	70
<b>Tabel 4.9</b> Pengujian Komunikasi Serial.....	71
<b>Tabel 4.10</b> Pengukuran Ketinggian Manual.....	86
<b>Tabel 4.11</b> Pengukuran Volume Manual.....	89

-----Halaman ini sengaja dikosongkan-----



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

PT. Pertamina Drilling Services Indonesia (PT. PDSI) merupakan perusahaan yang bergerak dalam bidang eksplorasi dan eksploitasi pengeboran minyak dan gas bumi, serta panas bumi. Selain itu, juga memberikan pelayanan *service* pengeboran yang terintegrasi. Minyak bumi merupakan sumber daya mineral yang banyak di temukan di dalam dasar bumi. Cara untuk mendapatkan minyak bumi tersebut yaitu dengan cara proses pengeboran. Dalam proses pengeboran terdapat beberapa sistem yang penting dalam melakukan kegiatan tersebut yaitu sistem tenaga (*power system*), sistem pengangkat (*hoisting system*), sistem pemutar (*rotating system*), sistem sirkulasi lumpur (*mud circulating system*) dan sistem pencegahan semburan (*blow out preventer system*). Sistem sirkulasi lumpur (*mud circulating system*) merupakan proses mengalirkan lumpur pemboran dari peralatan sirkulasi menuju ke rangkaian pipa bor. Saat proses pengeboran berlangsung sistem sirkulasi lumpur merupakan sistem yang sangat penting dikarenakan pada sistem ini berfungsi untuk menahan fluida formasi (tekanan formasi), mengangkat *cutting* ke permukaan, dan lain-lain [1].

Pada proses ini, pengisian lumpur pada lubang pengeboran harus sesuai dengan ukuran lubang dan juga harus sesuai dengan kondisi tanah yang telah dibor. Sebelum dialirkan menuju lubang bor, lumpur tersebut tersimpan dalam beberapa tangki, dimana dalam masing-masing tangki terdapat perlakuan yang berbeda, sehingga mempunyai sebutan berbeda dari masing-masing tangki, seperti *trip tank*, *satling tank*, *active tank*. Setiap tangki tersebut membutuhkan pengawasan volume lumpur terhadap keberlangsungan proses pengeboran dikarenakan keadaan volume lumpur merupakan bagian penting sebelum dimasukkan ke dalam lubang pengeboran. Pada kenyataan yang terdapat di lapangan saat pengukuran volume lumpur tangki hanya mengandalkan pandangan mata sehingga menyebabkan

ketidakakuratan pengukuran volume lumpur yang terdapat pada tangki. Hal tersebut dapat menghambat keadaan operasional pengeboran apabila saat pengeboran terjadi kekurangan stok lumpur [2]. Maka dari itu, dibutuhkan suatu alat *Pit Volume Totalizer* (PVT) yang dapat mengukur keadaan volume lumpur. Alat ini terdiri dari sensor pelampung yang diintegrasikan dengan mikrokontroler ATmega16 sebagai kontroler dan HMI sebagai *monitoring* keadaan volume lumpur. Dengan demikian, sistem *monitoring* volume lumpur telah mampu mengukur ketinggian dan volume lumpur dengan tingkat keakuratan  $\pm 99,15\%$ , sehingga kesalahan pengukuran dapat terminimalisasi.

## 1.2 Permasalahan

Pengisian lumpur yang terdapat pada *mud tank* akan terus mengisi dengan perlakuan lumpur yang berbeda-beda pada setiap tangki penampungan lumpur. Ketika proses pengeboran berlangsung masing-masing tangki membutuhkan pengawasan ketinggian dan volume lumpur guna mengetahui berapa volume lumpur yang dibutuhkan untuk sistem sirkulasi lumpur agar tidak terjadi kelebihan dan kekurangan lumpur saat dimasukkan kedalam lubang pengeboran.

## 1.3 Batasan Masalah

Pada penelitian ini diperlukan adanya pembatasan masalah yang digunakan untuk mempermudah pelaksanaan maupun penulisan tugas akhir sehing tidak menyimpang dari judul tugas akhir. Lingkup pembatasan masalah dalam laporan tugas akhir ini dibatasi pada :

- Sistem yang dibuat menggunakan mikrokontroler ATmega16 dan menggunakan pemrograman Bahasa C pada software CodeVision AVR.
- Menggunakan sensor potensiometer multiturn 10 kali.
- Sistem ini dirancang menggunakan pelampung.
- Media pengukuran menggunakan air.

## 1.4 Tujuan

Penelitian ini bertujuan untuk pengawasan keadaan volume lumpur dan ketinggian lumpur yang akan dimasukkan dalam lubang pengeboran, sehingga dapat meminimalisasi keadaan kelebihan dan kekurangan lumpur untuk sirkulasi pada saat proses pengeboran berlangsung.

## 1.5 Metodologi Penelitian

Dalam pelaksanaan tugas akhir dengan judul Rancang Bangun Sistem *Monitoring* Volume Lumpur pada *Mud Tank* Berbasis Mikrokontroler. Terdapat beberapa tahap yang perlu dipersiapkan yaitu tahap persiapan, tahap perancangan, tahap pembuatan, tahap pengujian dan analisis, serta tahap penyusunan laporan.

Pada tahap persiapan ini dilakukann studi literatur mengenai materi-materi yang memiliki hubungan dengan topik penelitian. Materi literatur ini dapat dicari melalui internet, ruang baca, perpustakaan, perpustakaan daerah, jurnal pada bidang tersebut dan penelitian-penelitian yang telah ada sebelumnya. Dalam tahapan ini juga dilakukan konsultasi dan diskusi dengan pembimbing yang berada di perusahaan serta pembimbing yang terdapat di kampus. Yang mempunyai tujuan untuk meluruskan jika terjadi pemahaman materi yang kurang sesuai dengan keadaan yang dilapangan serta dapat memberikan arahan, rekomendasi atau saran yang membangun untuk menunjang tugas akhir ini. Dengan demikian mahasiswa dapat menyelesaikan suatu permasalahan yang terjadi selama pengerjaan tugas akhir.

Tahap selanjutnya yaitu dilakukan tahap perancangan alat, langkah yang dilakukan yaitu perancangan sistem, yang meliputi perangkat keras mekanik, perangkat keras elektrik, dan perangkat lunak. Dalam perancangan keras mekanik terdiri dari desain mekanik sensor ketinggian (*level*) dan desain panel kontrol. Pada perancangan perangkat keras elektrik meliputi desain rangkaian power supply, desain rangkaian penguat, desain sistem minimum Atmega16, serta desain komunikasi

serial RS485. Sedangkan pada perancangan perangkat lunak meliputi dari desain tampilan *login channel*, tampilan desain *monitoring channel*, tampilan desain grafik dan database pada *Human Machine Interface* (HMI). Setelah pada tahap tersebut, dilakukan pembuatan alat atau implementasi dari perancangan tersebut.

Dalam pembuatan alat tersebut terdapat beberapa komponen untuk menunjang sistem tersebut, yaitu potensiometer, mikrokontroler Atmega16, Op-Amp IC LM741, modul komunikasi RS485. Serta implementasi HMI yang telah dirancang sebelumnya.

Selanjutnya dilakukan pada tahap pengujian dan analisis. Dari hasil pembuatan sistem tersebut akan dilakukan pengujian pada sistem yang telah dibuat meliputi pengujian dari perangkat keras mekanik, perangkat keras elektrik, dan perangkat lunak. Kemudian dari pengujian tersebut dilakukan analisis dari data yang telah diperoleh saat pengujian, serta membandingkan dengan hasil nyata dengan teori yang telah di pelajari sebelumnya.

Tahapan yang terakhir yaitu penyusunan laporan tugas akhir yang bertujuan untuk menyampaikan hasil dari kegiatan yang telah dikerjakan secara tertulis dan sebagai bukti pertanggung jawaban atas terselesaikan suatu penelitian yang telah dilakukan. Diharapkan buku ini dapat bermanfaat bagi pembaca dan bisa dijadikan pedoman dalam melanjutkan pengembangan ide ini.

## **1.6 Sistematika Laporan**

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

### **BABI PENDAHULUAN**

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

### **BAB II MONITORING VOLUME LUMPUR**

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari Sistem Sirkulasi Lumpur, Pengukuran Ketinggian dan Volume, Mikrokontroler Atmega16, Sensor Ketinggian, Pengondisian Sinyal *Analog*, Rangkaian Jembatan *Wheatstone*, IC LM741, Catu daya atau *Power supply*, Komunikasi Serial RS485, *Visual Basic*, *Codevision AVR*, *Database MySQL*.

### **BAB III PERANCANGAN *MONITORING* VOLUME LUMPUR**

Bab ini membahas tentang cara pembuatan alat baik secara mekanis, listrik, dan perangkat lunaknya.

### **BAB IV HASIL PENGUJIAN SISTEM *MONITORING* VOLUME LUMPUR**

Bab ini memuat tentang pemaparan dan analisis hasil pengujian alat pada keadaan sebenarnya. Seperti pengujian aktivitas motor DC, *power supply* dan sensor pendeteksi *level*. Pada tiap pengujian akan ada analisis terkait metode yang digunakan.

### **BAB V PENUTUP**

Bab ini berisi kesimpulan dan saran dari hasil pengambilan data yang telah diperoleh.

-----Halaman ini sengaja dikosongkan-----

## **BAB II**

### ***MONITORING VOLUME LUMPUR***

Bab ini membahas mengenai teori – teori yang berkaitan dengan peralatan yang akan dirancang. Teori yang mendukung penyelesaian tugas akhir ini diantaranya adalah mengenai Sistem Sirkulasi Lumpur (*Mud Circulating System*) Potensiometer *Multiturn*, Pengondisian Sinyal *Analog* (IC LM741), *Power supply* 5V, 9V dan -9V DC, Mikrokontroler ATmega16, Komunikasi Serial RS485, *Visual Basic*, *CodeVision AVR*, dan *Database MySQL*.

#### **2.1 Sistem Sirkulasi Lumpur [2]**

Suatu proses mengalirkan lumpur pemboran dari peralatan sirkulasi menuju ke rangkaian pipa bor. Sistem sirkulasi merupakan salah satu sistem yang memegang peranan penting di dalam operasi pengeboran putar (*rotary drilling*). Tugas utamanya adalah membantu sistem pemutar didalam “mengebor sumur” dengan menyediakan perlengkapan - perlengkapan yang sesuai untuk mengatur bahan-bahan lumpur dan tempat-tempat kerja untuk mempersiapkan, merawat dan mengganti fluida pengeboran.

Sistem sirkulasi tersusun oleh empat sub komponen utama, yaitu :

- Lumpur pengeboran (*drilling fluid*)
- Tempat persiapan (*preparation area*)
- Peralatan sirkulasi (*circulating equipment*)
- Tempat pengondisian lumpur (*conditioning area atau solid control equipment*)

Lumpur pengeboran (*drilling fluid*) merupakan faktor yang penting dalam pengeboran. Lumpur pengeboran pada mulanya hanya berfungsi sebagai pembawa serbuk bor (*cutting*) dari dasar lubang bor ke permukaan. Lumpur pengeboran mempunyai fungsi penting dalam operasi pengeboran, antara lain mengangkat *cutting* ke permukaan, mengontrol tekanan formasi, mendinginkan dan melumasi *bit* dan *drill string*, menahan *cutting* saat sirkulasi dihentikan, mengurangi sebagian

berat rangkaian pipa bor (*Bouyancy Effect*), melepas *cutting* dan pasir dipermukaan, mendapatkan informasi (*mud logging, sample log*), sebagai media *logging*.

## 2.2 Pengukuran Ketinggian Permukaan dan Volume [3]

Instrumentasi merupakan suatu alat yang sangat penting dalam suatu sistem pengukuran yang salah satunya pengukuran besarnya tinggi permukaan cairan, alat ini harus dapat berfungsi dengan baik sesuai dengan kebutuhan instrumentasi di industri. Alat instrumentasi ini merupakan salah satu faktor penting yang menentukan hasil produksi, dimana alat instrumentasi tersebut yang mengukur, mengontrol, mendeteksi, menganalisa, baik secara manual maupun secara otomatis.

Pengukuran dilakukan dengan tujuan untuk mendapatkan informasi yang akurat tentang suatu objek yang diukur. Pengukuran permukaan, volume, dan berat cairan dan bahan kering sering sekali dijumpai. Pengukuran yang teliti seringkali sulit didapatkan sehingga membutuhkan alat dengan ketelitian yang cukup tinggi. Metode umum yang digunakan untuk melaksanakan pengukuran dapat dilakukan dengan dua cara yaitu metoda langsung dan metoda mekanik. Metoda langsung dengan melihat tinggi permukaan cairan dan diduga kedalamannya dan ditunjukkan dalam satuan pengukuran panjang (meter). Dengan diketahuinya tinggi permukaan cairan maka volume dari cairan yang diukur dapat dicari bila dikehendaki. Sementara untuk metoda mekanik yaitu dengan menggunakan gaya pada cairan yang menghasilkan gerak mekanik. Pergerakan mekanik ini kemudian dikalibrasi kedalam bentuk skala angka-angka.

Terdapat beberapa faktor yang perlu dipertimbangkan dalam pemilihan alat ukur ketinggian :

- a. Batas ketinggian.
- b. Karakteristik fluida
  - Suhu
  - Tekanan
  - Gravitasi spesifikasi



- Apakah fluida bersih atau kotor, mengandung uap atau bahan padat.

c. Efek korosif

Beberapa jenis metode pengukuran level atau tinggi permukaan untuk fluida yang sering digunakan dalam proses industri, dapat dikelompokkan sebagai berikut :

- a. Tipe Pemindahan (*Displacement Type*).
- b. Tipe Perbedaan Tekanan (*Differential Pressure Type*).
- c. Tipe kapasitansi (*Capacitance type*)
- d. Tipe gelombang ultrasonik (*Ultrasonic type*)
- e. *Radar type*
- f. *Radiation type*

Penelitian ini menggunakan pengukuran ketinggian dengan tipe pemindahan atau *displacement* yang mempunyai prinsip kerja alat yaitu menggunakan pelampung yang diapungkan pada permukaan fluida, maka pelampung tersebut akan naik dan turun sesuai dengan arah gerakan dari permukaan fluida. Adapun kelebihan dan kekurangan dari tipe *displacement*, yaitu

Kelebihan :

- Mudah dalam pengoperasian
- Handal pada berbagai jenis *liquid* / cairan yang tenang, ataupun bergelembung.
- Dapat digunakan untuk mengukur *liquid interface*.

Kekurangan :

- Sistem mekanik yang rumit.
- *Range* terbatas, jangkauan yang lebih tinggi sulit diukur.
- Nilai pengukuran mudah tidak stabil.

Dari pengukuran ketinggian yang telah dilakukan, data tersebut di konversi menjadi data volume satuan barel. Dari data ketinggian maka dikalikan dengan luas dari permukaan media. Berikut merupakan perhitungan konversi dari ketinggian menjadi volume.

$$Volume (bbls) = \frac{(Luas) \times Tinggi}{158,9873} \times 1000 \dots \dots \dots (2.1)$$

Keterangan :    Luas = Media yang dipakai dalam pengukuran (m<sup>2</sup>)  
                  Tinggi = Ketinggian (m)

### 2.3 Mikrokontroler ATmega16 [4]

Mikrokontroler adalah sebuah sistem komputer lengkap dalam satu chip. Mikrokontroler lebih dari sekedar sebuah mikroprosesor karena sudah terdapat atau berisikan ROM (*Read-Only Memory*), RAM (*Read-Write Memory*), beberapa port masukan maupun keluaran, ADC (*Analog to Digital converter*), DAC (*Digital to Analog converter*) dan serial komunikasi.

Mikrokontroler AVR (*Alf and Vegard's Risc processor*) merupakan pengontrolan utama standar industri dan riset saat ini. Hal ini dikarenakan berbagai kelebihan yang dimilikinya dibandingkan mikroprosesor antara lain lebih murah, dukungan *software* dan dokumentasi yang memadai dan memerlukan komponen pendukung yang sangat sedikit. Salah satu tipe mikrokontroler AVR untuk aplikasi standar yang memiliki fitur memuaskan ialah ATmega16.

Mikrokontroler AVR standar memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode 16 bit. Secara internal mikrokontroler ATmega16 terdiri atas unit-unit fungsionalnya ALU (*Arithmetic and Logical Unit*), himpunan register kerja, register dan *decoder* instruksi, dan pewaktu beserta komponen kendali lainnya.

AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set Komputer*). Hampir semua instruksi dieksekusi dalam satu siklus *clock*. AVR mempunyai 32 register *general-purpose*, timer/counter fleksibel dengan mode *compare*, *interrupt* internal dan eksternal, serial UART, *programmable Watchdog Timer*, dan *mode power saving*, ADC dan PWM internal. AVR juga mempunyai *InSystem Programmable Flash on-chip* yang memungkinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI. ATmega16. ATmega16 mempunyai *throughput* mendekati 1 MIPS per MHz membuat desainer sistem untuk mengoptimasi konsumsi daya versus kecepatan proses.

### 2.3.1 Arsitektur ATMEGA16

Mikrokontroler ini menggunakan arsitektur Harvard yang memisahkan memori program dari memori data, baik port alamat maupun port data, sehingga saat mengakses program dan data dapat dilakukan secara bersamaan. Terdapat fitur-fitur yang dimiliki oleh ATmega16 sebagai berikut :

- Mikrokontroler AVR 8 Bit yang memiliki kemampuan tinggi, dengan daya rendah.
- Memiliki kapasitas Flash memori 16 KByte.
- Saluran I/O sebanyak 32 buah, yaitu Port A, Port B, Port C dan Port D
- CPU terdiri atas 32 register.
- Unit Interupsi *Internal* dan *Eksternal*.
- ADC *internal* dengan 10 bit 8 *channel*.
- Sistem mikroprosesor 8 bit berbasis RISC dengan kecepatan maksimal 16 MHz.
- Port USART untuk komunikasi serial.

Dengan fitur-fitur seperti diatas, pembuatan alat menggunakan ATmega16 menjadi lebih sederhana dan tidak memerlukan IC pendukung yang banyak.

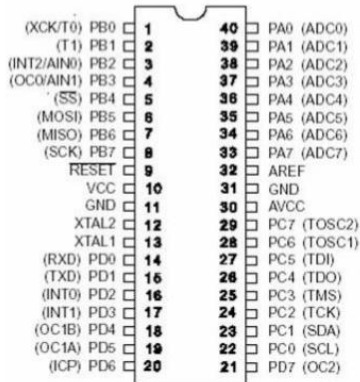
### 2.3.2 Konfigurasi Pin ATMEGA16

Konfigurasi pin mikrokontroler ATmega16 dengan kemasan 40 pin dapat dilihat pada Gambar 2.1. ATmega16 memiliki 32 pin yang digunakan untuk *input/output*, pin-pin tersebut terdiri dari 8 pin di setiap *port* mikrokontroler. Dalam komunikasi serial, maka hanya *port* D yang dapat digunakan karena fungsi khusus yang dimilikinya Untuk lebih jelas akan ditunjukkan pada Tabel-Tabel fungsi khusus *port*.

Berikut ini adalah penjelasan umum susunan kaki dari ATmega16:

- VCC.

Merupakan pin masukan positif catu daya. Setiap peranti elektronika digital membutuhkan sumber daya yang umumnya sebesar 5V.



**Gambar 2.1** Susunan Kaki Pin ATmega16

- **GND**

Merupakan pin masukan ground yang digunakan sebagai pengaman rangkaian dan sebagai referensi tegangan nol pemasok tegangan digital.

- **Port A (PA0 - PA7).**

Merupakan pin I/O dua arah dan dapat diprogram sebagai pin masukan ADC. *Port A* berfungsi sebagai *input analog* pada konverter A/D. *Port A* juga sebagai suatu *port I/O* 8-bit dua arah, jika A/D konverter tidak digunakan. Pin-pin *port* dapat menyediakan resistor internal *pull-up* (yang dipilih untuk masing-masing bit). *Port A output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Ketika pin PA0 ke PA7 digunakan sebagai *input* dan secara eksternal ditarik rendah, pin-pin akan memungkinkan arus sumber jika resistor internal *pull-up* diaktifkan. Pin *port A* adalah *tri-stated* manakala suatu kondisi *reset* menjadi aktif, sekalipun waktu habis.

**Tabel 2.1** Fungsi Khusus *Port A* Atmega 16

<b>Port Pin</b>	<b>Alternate Function</b>
PA7	ADC7 : <i>Analog Digital Converter</i> ke 7
PA6	ADC6 : <i>Analog Digital Converter</i> ke 6
PA5	ADC5 : <i>Analog Digital Converter</i> ke 5

**Tabel 2.1** Fungsi Khusus *Port A* Atmega 16 (lanjutan)

<b>Port Pin</b>	<b>Alternate Function</b>
PA4	ADC4 : <i>Analog Digital Converter</i> ke 4
PA3	ADC3 : <i>Analog Digital Converter</i> ke 3
PA2	ADC2 : <i>Analog Digital Converter</i> ke 2
PA1	ADC1 : <i>Analog Digital Converter</i> ke 1
PA0	ADC0 : <i>Analog Digital Converter</i> ke 0

- *Port B* (PB0 - PB7).

Merupakan pin I/O dua arah dan pin fungsi khusus, yaitu *Timer/Counter*, komparator *analog* dan SPI. *Port B* adalah suatu *port* I/O 8-bit dua arah dengan resistor internal *pull-up* (yang dipilih untuk beberapa bit). *Port B output buffer* mempunyai karakteristik gerakan simetris dengan keduanya sink tinggi dan kemampuan sumber. Sebagai *input*, pin *port B* yang secara eksternal ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pin *port B* adalah *tri-stated* manakala suatu kondisi *reset* menjadi aktif, sekalipun waktu habis.

**Tabel 2.2** Fungsi Khusus *Port B* Atmega 16

<b>Port Pin</b>	<b>Alternate Function</b>
PB7	SCK : SPI serial waktu
PB6	MISO : SPI master <i>input</i> / <i>slave output</i>
PB5	MOSI : SPI master <i>output</i> / <i>slave input</i>
PB4	SS : SPI <i>slave select input</i>
PB3	AIN1 : pembanding <i>analog, input negative</i> OC0 : <i>Timer / counter 0 output</i> (pembanding <i>output</i> )
PB2	AIN0 : Pembanding <i>analog, input positive</i> INT2 : External Interrupt

<b>Port Pin</b>	<b>Alternate Function</b>
PB1	T1 : <i>Timer / Counter 1 eksternal counter output</i>
PB0	T0 : <i>Timer / Counter 0 eksternal counter output</i> XCK : <i>USART waktu eksternal input / output</i>

- *Port C (PC0 - PC7).*

Merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator TWI, komparator *analog* dan *timer* osilator. *Port C output buffer* mempunyai karakteristik gerakan simetris dengan keduanya sink tinggi dan kemampuan sumber. Sebagai *input*, pin bandar C yang secara eksternal ditarik rendah akan arus sumber jika resistor pull-up diaktifkan. Pin *port C* adalah *tri-stated* manakala suatu kondisi *reset* menjadi aktif, sekalipun waktu habis.

**Tabel 2.3** Fungsi Khusus *Port C* Atmega 16

<b>Port Pin</b>	<b>Alternate Function</b>
PC7	TOSC2 : Waktu <i>Oscillator</i> Pin 2
PC6	TOSC1 : Waktu <i>Oscillator</i> Pin 1
PC5	TDI : <i>JTAG Test Data Input</i>
PC4	TDO : <i>JTAG Test Data Output</i>
PC3	TMS : <i>JTAG Test Mode Select</i>
PC2	TCK : <i>Test Clock</i>
PC1	SDA : Dua Penghubung Serial Data <i>I/O</i>
PC0	SCL : Dua Penghubung Serial Waktu

- *Port D (PD0 - PD7).*

Merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator *analog*, interupsi eksternal dan komunikasi serial. *Port D* adalah suatu *port I/O* 8-bit dua arah dengan resistor internal *pull-up* (yang dipilih untuk beberapa bit). *Port D output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Sebagai *input*, pin *port D* yang secara eksternal ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan.

**Tabel 2.4** Fungsi Khusus *Port D* Atmega 16

<b>Port Pin</b>	<b>Alternate Function</b>
PD7	OC2 : <i>Timer / Counter 2 Output</i> (Hasil <i>Output</i> )

**Tabel 2.4** Fungsi Khusus *Port D* Atmega 16 (lanjutan)

<b>Port Pin</b>	<b>Alternate Function</b>
PD6	ICP1 : <i>Timer / Counter 1 Input Pin</i>
PD5	OC1A : <i>Timer / Counter 1 Hasil Output A</i>
PD4	OC1B : <i>Timer / Counter 1 Hasil Output B</i>
PD3	INT1 : <i>External Interrupt 1 Input</i>
PD2	INT0 : <i>External Interrupt 0 Output</i>
PD1	TXD : <i>USART Output Pin</i>
PD0	RXD : <i>USART Input Pin</i>

- *Reset*  
Merupakan pin yang digunakan untuk *me-reset ulang* mikrokontroler agar membaca dari awal kembali.
- XTAL 1 dan XTAL 2  
Sebagai pin masukan *clock* eksternal. Suatu mikrokontroler membutuhkan sumber detak (*clock*) agar dapat mengeksekusi instruksi yang ada di memori. Semakin tinggi nilai kristalnya maka semakin cepat mikrokontroler tersebut,
- AVCC dan AREF  
Fungsi dari AVCC sebagai pin masukan tegangan untuk ADC, sedangkan fungsi dari AREF sebagai pin masukan tegangan referensi.

## 2.4 Sensor Ketinggian

Sensor adalah alat yang digunakan untuk mendeteksi dan sering berfungsi untuk mengukur magnitudo sesuatu. Sensor adalah transduser yang digunakan untuk mengubah variasi mekanis, magnetis, panas,

sinar dan kimia menjadi tegangan dan arus listrik. Sensor biasanya dikategorikan melalui pengukur dan memegang peranan penting dalam pengendalian proses parbikasi modern. Sensor memberikan ekivalen mata, pendengaran, hidung, lidah untuk menjadi otak mikroprosesor dari sistem otomatisasi industri.

Sensor yang digunakan untuk mengukur ketinggian yaitu menggunakan *precision potentiometer multiturn* dengan nilai tahanan yang dapat berubah-ubah. Potensiometer merupakan resistor yang resistansinya dapat berubah ubah. Berdasarkan bentuknya potensiometer ada 3 jenis yaitu potensiometer *slider*, *rotary*, dan *trimmer*. Potensiometer rotari merupakan jenis potensiometer yang resistansinya berubah ketika asnya diputar.

Dua pin pada potensiometer membentuk sebuah jalur elemen resistif dengan *wiper* sebagai penentu rasio resistansinya. Dengan sifat perubahan resistansi tersebut potensiometer seringkali digunakan sebagai *input* maupun sensor. Potensiometer harus digunakan secara hati- hati. Apabila posisi nilai tahanan menjadi sangat kecil, maka arus listrik yang melewati bagian jalur itu menjadi terlalu besar dan dapat menghancurkan jalur karbon. Oleh karena itu, pasang resistor berukuran kecil secara seri terhadap jalur data, guna membatasi arus listrik agar tetap berada pada taraf aman. Kekurangan menggunakan potensiometer adalah tidak stabil, dan mungkin tidak mempunyai penguraian yang baik yaitu tidak dapat diset pada satu nilai yang tepat. Keluaran sensor potensiometer ini adalah berupa tegangan dengan persamaan 2.2 berikut ini:

$$V(output) = \frac{R1}{R2} x V(Input) \dots\dots\dots(2.2)$$

Keterangan:

- a. R1 = perubahan hambatan.
- b. R2 = hambatan maksimum.

Dari persamaan (2.2) dapat diamati bahwa tegangan  $V(output)$  berbanding lurus dengan perubahan nilai hambatan, perubahan



tegangan V (*output*) inilah yang nantinya akan digunakan sebagai indikator perubahan level ketinggian air. Atas dasar itulah maka digunakanlah potensiometer sebagai sensor pendeteksi level ketinggian air. Gambar 2.2 merupakan tampilan dari komponen *rotary* potensiometer yang digunakan pada penelitian ini.



**Gambar 2.2** Sensor Potensiometer

## 2.5 Pengondisian Sinyal *Analog* [5]

Pengondisian sinyal *analog* merupakan suatu konversi sinyal *analog* menjadi bentuk yang lebih sesuai yang merupakan antarmuka dengan elemen-elemen lain dalam suatu kontrol proses. Sebuah transduser mengukur suatu variabel dinamik dengan mengkonversinya kedalam sinyal elektrik. Untuk mengembangkan transduser seperti ini, banyak dipengaruhi oleh kondisi alam sehingga hanya ada beberapa tipe yang dapat digunakan untuk mendapatkan hasil yang sesuai.

Efek pengkondisi sinyal sering dinyatakan dengan fungsi alihnya (*transfer function*). Dengan istilah ini kita menghubungkan efek yang ditimbulkan dengan sinyal *input*. Jadi, sebuah amplifier sederhana mempunyai fungsi alih dari beberapa konstanta yang, ketika dikalikan dengan tegangan *input*, memberikan tegangan *output*. Terdapat beberapa macam cara untuk mengkondisikan sinyal *analog*, yaitu :

- Perubahan level sinyal.
- Linierisasi.
- Konversi.
- Penyaringan dan Penyepadan Impedansi

Rangkaian penguat ini menggunakan perubahan level dan sinyal dengan tujuan untuk menguatkan tegangan yang dihasilkan oleh sensor potensiometer.

### **2.5.1 Perubahan Level Sinyal**

Mengkondisikan sinyal *analog* yang diperlukan dengan pertama mengubah menjadi nol ketika keluaran sensor adalah 0,2 V. Perlakuan tersebut dapat dilakukan dengan mengurangi 0,2 dan keluaran sensor, yang disebut pergeseran nol atau penyesuaian bias. Saat ini mempunyai tegangan antara 0 V sampai 0,4 V, sehingga memerlukan tegangan yang lebih besar. Jika dikalikan dengan tegangan 12,5, tegangan keluaran yang baru akan bervariasi antara 0 V sampai 5 V seperti yang diperlukan. Hal ini disebut penguatan dan 12,5 disebut perbesaran (*gain*). Dalam kasus lain kita perlu membuat keluaran sensor menjadi lebih kecil yang disebut dengan pelemahan. Dalam mendesain bias dan rangkaian penguat, harus memperhatikan beberapa faktor antara lain tanggap frekuensi, impedansi keluaran, dan impedansi masukan.

## **2.6 Rangkaian Jembatan *Wheatstone* dan Potensiometer [6]**

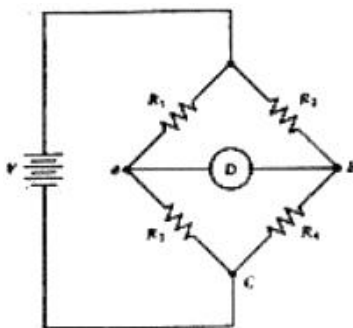
Rangkaian jembatan *wheatstone* terutama digunakan sebagai sebuah alat pengukur perubahan tahanan yang akurat. Rangkaian seperti ini terutama berguna bila perubahan fraksional dalam impedansi sangat kecil. Rangkaian potensiometer digunakan untuk mengukur tegangan dengan akurasi yang baik dan impedansi sangat tinggi.

### **2.6.1 Rangkaian Jembatan *Wheatstone***

Rangkaian jembatan *wheatstone* adalah rangkaian pasif yang digunakan untuk mengukur impedansi dengan teknik penyesuaian potensial. Dalam rangkaian ini, seperangkat impedansi yang telah diketahui secara akurat diatur nilainya dalam hubungannya terhadap satu yang belum diketahui sampai suatu kondisi yang ada dimana perbedaan potensial antara dua titik dalam rangkaian adalah nol, yaitu

setimbang. Kondisi ini menetapkan sebuah persamaan yang digunakan untuk menemukan impedansi yang tidak diketahui berkenaan dengan nilai-nilai yang diketahui.

Rangkaian jembatan yang paling sederhana dan paling umum adalah jembatan *DC Wheatstone* seperti diperlihatkan pada Gambar 2.3. Rangkaian ini digunakan dalam aplikasi pengkondisi sinyal dimana transduser mengubah tahanan dengan perubahan variabel dinamik. Beberapa modifikasi dari jembatan dasar ini juga dipakai untuk aplikasi spesifik lainnya. Pada Gambar 2.3 obyek yang diberi label D adalah detektor setimbang yang digunakan untuk membandingkan potensial titik a dan b dari rangkaian. Dalam aplikasi paling modern detektor setimbang adalah amplifier diferensial impedansi *input* sangat tinggi. Dalam beberapa kasus, Galvanometer yang sensitif dengan impedansi yang relatif rendah bisa digunakan, khususnya untuk kalibrasi atau instrumen-instrumen pengukuran tunggal.



**Gambar 2.3** Jembatan DC *Wheatstone*

Dalam kasus ini beda potensial,  $\Delta V$  antara titik a dan b, adalah

$$\Delta V = V_a - V_b \dots \dots \dots (2.3)$$

Dimana

- $V_a$  = potensial titik a terhadap c
- $V_b$  = potensial titik b terhadap c

Nilai  $V_a$  dan  $V_b$  sekarang dapat dicari dengan memperhatikan bahwa  $V_a$  adalah hanya tegangan sumber,  $V$ , dibagi antara  $R_1$  dan  $R_3$

$$V_a = \frac{VR_3}{R_1 + R_3} \dots\dots\dots(2.4)$$

Dengan cara yang sama  $V_b$  adalah tegangan yang terbagi diberikan oleh

$$V_b = \frac{VR_4}{R_2 + R_4} \dots\dots\dots(2.5)$$

Dimana

- $V$  = tegangan sumber jembatan
- $R_1, R_2, R_3, R_4$  = resistor-resistor jembatan seperti diberikan oleh Gambar 2.3.

Jika sekarang dikombinasikan Persamaan (2.3), (2.4), (2.5), beda tegangan atau *offset* tegangan, dapat ditulis

$$\Delta V = \frac{VR_3}{R_1 + R_3} - \frac{VR_4}{R_2 + R_4} \dots\dots\dots(2.6)$$

Setelah beberapa aljabar, pembaca dapat memperlihatkan bahwa persamaan ini berkurang menjadi

$$\Delta V = V \frac{R_2 R_3 - R_1 R_4}{(R_1 + R_3)(R_2 + R_4)} \dots\dots\dots(2.7)$$

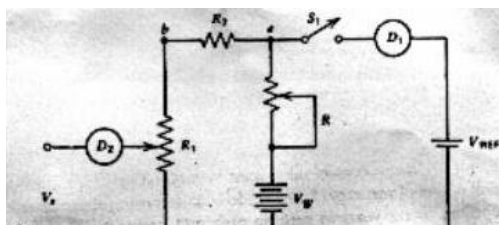
Persamaan (2.7) memperlihatkan bagaimana beda potensial melalui detektor adalah fungsi dari tegangan sumber dan nilai resistor. Karena tampilan yang berbeda dalam numerator Persamaan (6), jelas bahwa kombinasi khusus dari resistor dapat ditemukan yang akan menghasilkan perbedaan nol dan tegangan nol melewati detektor yaitu setimbang.

$$R_3R_2 = R_1R_4 \dots \dots \dots (2.8)$$

Persamaan (2.8) mengindikasikan bahwa kapan saja sebuah jembatan *Wheatstone* dipasang dan resistor diatur untuk setimbang detektor, nilai-nilai resistor harus memenuhi persamaan yang diindikasikan. Tidak masalah jika tegangan sumber berubah, kondisi setimbang dipertahankan. Persamaan tersebut menekankan aplikasi jembatan *Wheatstone* untuk aplikasi kontrol proses yang menggunakan detektor impedansi *input* tinggi.

### 2.6.2 Rangkaian Potensiometer

Pengukuran tegangan dalam kontrol proses sering kali harus dibuat pada impedansi sangat tinggi dan dengan tingkat akurasi yang tinggi. Banyak rangkaian modern yang menggunakan peralatan aktif telah dikembangkan pada akhir-akhir ini untuk melakukan pengukuran-pengukuran seperti ini. Selama bertahun-tahun metode yang dapat diandalkan untuk pengukuran-pengukuran seperti ini, yang akurat dan impedansi tinggi, hanya potensiometer. Pada dasarnya, rangkaian ini adalah sebuah pembagi tegangan yang mengukur tegangan yang tidak diketahui dengan mengatur yang telah diketahui, yaitu tegangan yang terbagi sampai sesuai/cocok dengan yang diketahui. Pembagi tegangan dikonstruksi oleh  $R_1$ ,  $R_2$  dan  $R$  secara seri yang dihubungkan ke tegangan sumber kerja.,  $V_w$ .  $R_2$  adalah resistor presisi dan tertentu, sedangkan  $R_1$  adalah resistor yang presisi dan



Gambar 2.4 Rangkaian Dasar Potensiometer

variabel linier. Resistor kalibrasi R adalah variabel (yang nilai sebenarnya belum pernah digunakan dalam perhitungan apa pun), dan  $V_w$  adalah sumber yang mempunyai tegangan yang memadai (seperti yang akan ditetapkan nanti) dan stabil. Supply  $V_{REF}$  adalah sebuah standar kalibrasi yang mempunyai tegangan yang telah diketahui secara akurat. Unit  $D_1$  dan  $D_2$  keduanya adalah detektor setimbang dan bisa berupa galvanometer ataupun detektor tegangan impedansi tinggi.  $V_x$  adalah tegangan yang tidak diketahui yang akan diukur.

Kalibrasi dari pembagi tegangan dipenuhi dengan menutup saklar  $S_1$  dan mengatur R sampai detektor  $D_1$  mengindikasikan setimbang. Dalam kondisi ini kita akan menetapkan/membuktikan bahwa  $V_a = V_{REF}$  sesuai akurasi dari detektor kesetimbangan. Secara efektif dapat mengkalibrasi rangkaian pembagi karena  $V_a$  dibagi antara resistor presisi  $R_1$  dan  $R_2$ . Penyapu  $R_1$  menyapu tegangan antara *zero* pada bagian bawah dan  $V_b$  pada bagian atas dari resistor variabel. Tegangan  $V_b$  dicari dari

$$V_b = \frac{R_1 V_a}{R_1 + R_2} \dots\dots\dots(2.9)$$

Karena  $V_a = V_{REF}$ , mempunyai identifikasi  $V_b$  secara langsung dalam hubungan  $V_{REF}$ . Sekarang jika penyapu  $R_1$  adalah bagian/pecahan  $\alpha$  dari sisi ground, tahanan diatas penyapu adalah  $(1-\alpha)R$ . Jika sebuah tegangan yang tidak diketahui diberikan sebagaimana diperlihatkan pada Gambar 2.4 dan penyapu diatur sampai detektor  $D_2$  menunjukkan nol, tegangan penyapu dan tegangan yang tidak diketahui adalah sama. Jadi, tegangan yang tidak diketahui diberikan oleh

$$V_x = \alpha V_b \dots\dots\dots(2.10)$$

Dimana

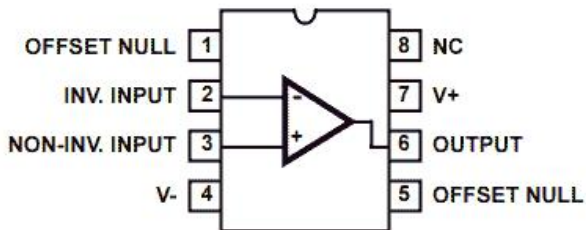
$\alpha$ = bagian/pecahan R untuk terjadinya kondisi setimbang

$V_b$  = tegangan titik b yang diberikan oleh Persamaan (2.9)

Dalam beberapa kasus resistor variabel  $R_1$  diberi penskalaan dengan pembagian, seperti pembagian yang dapat dibaca 1000. Dalam kasus ini,  $\alpha$  adalah hanya sejumlah pembagian yang menghasilkan keadaan setimbang dari detektor  $D_2$ . Perhatikan bahwa sekali pembagi dikalibrasi, tegangan acuan  $V_{REF}$  dan detektor  $D_1$  tidak diperlukan lebih lama.

## 2.7 IC LM741

Komponen ini merupakan operasional amplifier populer yang dikemas dalam bentuk dual in-line package (DIP). Kemasan IC jenis DIP memiliki tanda bulatan atau strip pada salah satu sudutnya untuk menandai arah pin atau kaki nomor 1 dari IC tersebut. Penomoran IC dalam kemasan DIP adalah berlawanan arah jarum jam dimulai dari pin yang terletak paling dekat dengan tanda bulat atau strip pada kemasan DIP tersebut.



Gambar 2.5 DIP 8 pin LM741

IC LM741 berisi satu buah Op-Amp, terdapat banyak tipe IC lain yang memiliki dua atau lebih Op-Amp dalam suatu kemasan DIP. IC Op-Amp memiliki karakteristik yang sangat mirip dengan konsep Op-Amp ideal pada analisis rangkaian. Pada IC Op-Amp terdapat batasan-batasan penting yang perlu diperhatikan. Pertama, tegangan maksimum *power supply* tidak boleh melebihi rating maksimum, karena akan merusak IC. Kedua, tegangan *output* dari IC op-amp biasanya satu atau

dua volt lebih kecil dari tegangan *power supply*. Sebagai contoh, tegangan swing *output* dari suatu op-amp dengan tegangan supply 15 V adalah  $\pm 13V$ . Ketiga, arus *output* dari sebagian besar op-amp memiliki batas pada 30mA, yang berarti bahwa resistansi beban yang ditambahkan pada *output* op-amp harus cukup besar sehingga pada tegangan *output* maksimum, arus *output* yang mengalir tidak melebihi batas arus maksimum.

IC LM741 memiliki kemasan DIP 8 pin seperti terlihat pada Gambar berikut:

Pada IC ini terdapat dua pin *input*, dua pin *power supply*, satu pin *output*, satu pin NC (*No Connection*), dan dua pin *offset null*. Pin *offset null* memungkinkan untuk melakukan sedikit pengaturan terhadap arus internal di dalam IC untuk memaksa tegangan *output* menjadi nol ketika kedua *input* bernilai nol.

Tujuan digunakan komponen IC LM741 pada penelitian ini yaitu digunakan sebagai pengondisian sinyal *analog* atau penguatan tegangan untuk sensor ketinggian agar nilai dari keluaran tegangan tersebut terbaca tegang dari 0 – 5 volt. Komponen ini bisa digunakan untuk sistem *analog* komputer, penguat video atau Gambar, penguat audio, osilator, *detector* dan lain-lain tergantung kebutuhan yang diinginkan.

## 2.8 Catu Daya atau *Power supply*

Catu daya atau sering disebut dengan *power supply* adalah perangkat elektronika yang berguna sebagai sumber daya untuk perangkat lain. Secara umum istilah catu daya berarti suatu sistem penyearah-filter yang mengubah ac menjadi dc murni. Sumber DC seringkali dapat menjalankan peralatan-peralatan elektronika secara langsung, meskipun mungkin diperlukan beberapa cara untuk meregulasi dan menjaga suatu ggl agar tetap meskipun beban berubah-ubah. Energi yang paling mudah tersedia adalah arus bolak-balik, harus diubah atau disearahkan menjadi dc berpulsa (*pulsating dc*), yang selanjutnya harus diratakan atau disaring menjadi tegangan yang tidak



berubah-ubah. Tegangan dc juga memerlukan regulasi tegangan agar dapat menjalankan rangkaian dengan sebaiknya.

Secara garis besar, pencatu daya listrik dibagi menjadi dua macam, yaitu pencatu daya tak distabilkan dan pencatu daya distabilkan. Pencatu daya tak distabilkan merupakan jenis pencatu daya yang paling sederhana. Pada pencatu daya jenis ini, tegangan maupun arus keluaran dari pencatu daya tidak distabilkan, sehingga berubah-ubah sesuai keadaan tegangan masukan dan beban pada keluaran. Pencatu daya jenis ini biasanya digunakan pada peranti elektronika sederhana yang tidak sensitif akan perubahan tegangan. Pencatu jenis ini juga banyak digunakan pada penguat daya tinggi untuk mengkompensasi lonjakan tegangan keluaran pada penguat.

Pencatu daya distabilkan pencatu jenis ini menggunakan suatu mekanisme lolos balik untuk menstabilkan tegangan keluarannya, bebas dari variasi tegangan masukan, beban keluaran, maupun dengung. Ada dua jenis yang digunakan untuk menstabilkan tegangan keluaran, antara lain:

- Pencatu Daya Linier.

Merupakan jenis pencatu daya yang umum digunakan. Cara kerja dari pencatu daya ini adalah mengubah tegangan AC menjadi tegangan AC lain yang lebih kecil dengan bantuan Transformator. Tegangan ini kemudian disearahkan dengan menggunakan rangkaian penyearah tegangan, dan di bagian akhir ditambahkan kondensator sebagai penghalus tegangan sehingga tegangan DC yang dihasilkan oleh pencatu daya jenis ini tidak terlalu bergelombang. Selain menggunakan diode sebagai penyearah, rangkaian lain dari jenis ini dapat menggunakan regulator tegangan linier sehingga tegangan yang dihasilkan lebih baik daripada rangkaian yang menggunakan dioda. Pencatu daya jenis ini biasanya dapat menghasilkan tegangan DC yang bervariasi antara 0 - 60 Volt dengan arus antara 0 - 10 Ampere.

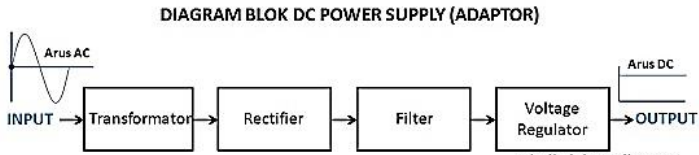
- Pencatu Daya Saklar,

Pencatu daya jenis ini menggunakan metode yang berbeda dengan pencatu daya linier. Pada jenis ini, tegangan AC yang masuk ke dalam rangkaian langsung disearahkan oleh rangkaian penyearah tanpa menggunakan bantuan transformer. Cara menyearahkan tegangan tersebut adalah dengan menggunakan frekuensi tinggi antara 10KHz hingga 1MHz, dimana frekuensi ini jauh lebih tinggi daripada frekuensi AC yang sekitar 50Hz. Pada pencatu daya sakelar biasanya diberikan rangkaian umpan balik agar tegangan dan arus yang keluar dari rangkaian ini dapat dikontrol dengan baik [7].

### **2.8.1 Prinsip Kerja DC Power supply [8]**

Arus Listrik yang kita gunakan di rumah, kantor dan pabrik pada umumnya adalah dibangkitkan, dikirim dan didistribusikan ke tempat masing-masing dalam bentuk Arus Bolak-balik atau arus AC (*Alternating Current*). Hal ini dikarenakan pembangkitan dan pendistribusian arus listrik melalui bentuk arus bolak-balik (AC) merupakan cara yang paling ekonomis dibandingkan dalam bentuk arus searah atau arus DC (*Direct Current*).

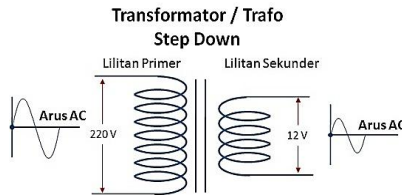
Akan tetapi, peralatan elektronika yang kita gunakan sekarang ini sebagian besar membutuhkan arus DC dengan tegangan yang lebih rendah untuk pengoperasiannya. Oleh karena itu, hampir setiap peralatan Elektronika memiliki sebuah rangkaian yang berfungsi untuk melakukan konversi arus listrik dari arus AC menjadi arus DC dan juga untuk menyediakan tegangan yang sesuai dengan rangkaian Elektronika-nya. Rangkaian yang mengubah arus listrik AC menjadi DC ini disebut dengan DC *Power supply* atau dalam bahasa Indonesia disebut dengan Catu daya DC. Sebuah DC *Power supply* atau Adaptor pada dasarnya memiliki 4 bagian utama agar dapat menghasilkan arus DC yang stabil. Keempat bagian utama tersebut diantaranya adalah *Transformer*, *Rectifier*, *Filter* dan *Voltage Regulator*. Sebelum kita membahas lebih lanjut mengenai Prinsip Kerja DC *Power supply*, sebaiknya kita mengetahui Blok-blok dasar yang membentuk sebuah DC *Power supply* atau Pencatu daya ini. Dibawah ini adalah Diagram Blok DC *Power supply* (Adaptor) pada umumnya.



**Gambar 2.6** Blok Diagram DC *Power supply*

a. Transformator (Transformer / Trafo)

Transformator (Transformer) atau disingkat dengan Trafo yang digunakan untuk DC *Power supply* adalah Transformer jenis *Step-down* yang berfungsi untuk menurunkan tegangan listrik sesuai dengan kebutuhan komponen Elektronika yang terdapat pada rangkaian adaptor (DC *Power supply*). Transformator bekerja berdasarkan prinsip



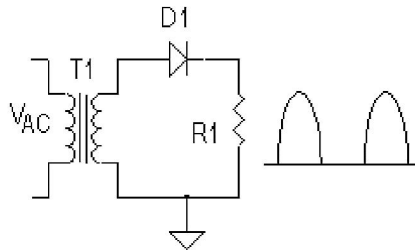
**Gambar 2.7** Transformator / Trafo Step Down

Induksi elektromagnetik yang terdiri dari 2 bagian utama yang berbentuk lilitan yaitu lilitan Primer dan lilitan Sekunder. Lilitan Primer merupakan *Input* dari pada Transformator sedangkan *Output*-nya adalah pada lilitan sekunder. Meskipun tegangan telah diturunkan, *Output* dari Transformator masih berbentuk arus bolak-balik (arus AC) yang harus diproses selanjutnya.

b. Penyearah Gelombang (*Rectifier*)

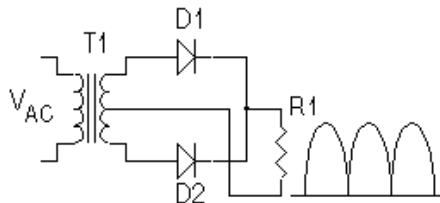
*Rectifier* atau penyearah gelombang adalah rangkaian Elektronika dalam *Power supply* (catu daya) yang berfungsi untuk mengubah gelombang AC menjadi gelombang DC setelah tegangannya

diturunkan oleh Transformator *Step down*. Rangkaian *Rectifier* biasanya terdiri dari komponen Dioda. Terdapat 2 jenis rangkaian *Rectifier* dalam *Power supply* yaitu “*Half Wave Rectifier*” yang hanya terdiri dari 1 komponen Dioda dan “*Full Wave Rectifier*” yang terdiri dari 2 atau 4 komponen dioda. Prinsip penyearah (*rectifier*) yang paling sederhana ditunjukkan pada Gambar 2.8. berikut ini. Transformator diperlukan untuk menurunkan tegangan AC dari jala-jala listrik pada kumparan primernya menjadi tegangan AC yang lebih kecil pada kumparan sekundernya.

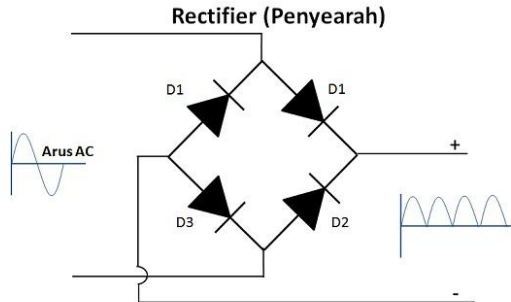


**Gambar 2.8** Rangkaian Penyearah Sederhana

Pada rangkaian Gambar 2.8, dioda berperan untuk meneruskan tegangan positif ke beban RL. Ini yang disebut dengan penyearah setengah gelombang (*half wave*). Untuk mendapatkan penyearah gelombang penuh (*full wave*) diperlukan transformator dengan *center tap* (CT) seperti pada Gambar 2.9.



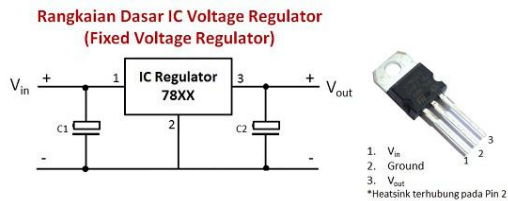
**Gambar 2.9** Rangkaian Penyearah Gelombang Penuh



**Gambar 2.10** Rangkaian Penyearah DC *Power supply*

c. Pengatur Tegangan (*Voltage Regulator*)

Untuk menghasilkan tegangan dan arus DC (arus searah) yang tetap dan stabil, diperlukan *voltage regulator* yang berfungsi untuk mengatur tegangan sehingga tegangan *output* tidak dipengaruhi oleh suhu, arus beban dan juga tegangan *input* yang berasal *output filter*. *voltage regulator* pada umumnya terdiri dari dioda zener, transistor



**Gambar 2.11** Rangkaian Dasar IC *Voltage Regulator*

atau IC (*Integrated Circuit*). Pada DC *power supply* yang canggih, biasanya *voltage regulator* juga dilengkapi dengan *short circuit protection* (perlindungan atas hubung singkat), *current limiting* (pembatas arus) ataupun *over voltage protection* (perlindungan atas kelebihan tegangan).

## 2.9 Komunikasi Serial RS485 [9]

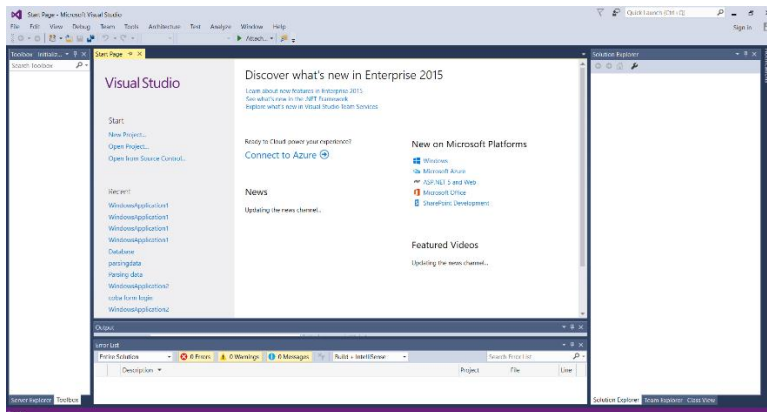
Saluran RS232 hanya dipakai untuk menghubungkan DTE dengan DCE dalam jarak pendek, untuk jarak lebih jauh bisa dipakai

saluran arus (*current loop*) tapi tidak untuk kecepatan transmisi tinggi. RS485 bisa dipakai untuk saluran sampai sejauh 4000 feet dan kecepatan lebih dari 1 Megabit/detik. *Standard* RS485 ditetapkan oleh *Electronic Industry Association* dan *Telecommunication Industry Association* pada tahun 1983. Nama lengkapnya adalah *EIA/TIA-485 Standard for Electrical Characteristics of Generators and Receivers for use in a Balanced Digital Multipoint System*.

Standard RS485 hanya membicarakan karakteristik sinyal dalam transmisi data secara *Balanced Digital Multipoint System*, jadi jauh lebih sederhana dibanding dengan *standard* RS232 yang mencakup ketentuan tentang karakteristik sinyal, macam-macam sinyal dan konektor yang dipakai, serta konfigurasi sinyal pada kaki-kaki di konektor dan juga penentuan tata cara pertukaran informasi antara komputer dan alat-alat pelengkapya..

## 2.10 Visual Basic .NET [10]

Bahasa pemrograman komputer visual basic merupakan perintah-perintah atau instruksi yang dimengerti oleh komputer untuk menghasilkan program-program aplikasi berbasis windows.



Gambar 2.12 Tampilan Awal Visual Basic

*Visual Basic* memungkinkan pembuatan aplikasi *Graphical User Interface* (GUI) atau pemrograman yang menggunakan tampilan grafis sebagai alat komunikasi dengan pemakainya. Dalam *Visual Basic* untuk pembuatan tampilan *user interface* relatif mudah dilakukan karena hanya perlu meletakkan objek-objek grafis ke sumber (*form*) yang sudah disediakan *Visual Basic*. Setelah itu cukup mengatur properti dari objek-objek tersebut.

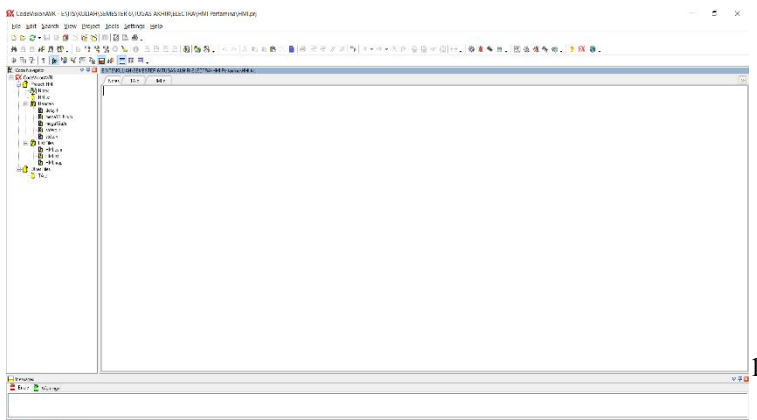
Beberapa kemampuan atau manfaat dari *Visual Basic* diantaranya adalah :

- Untuk membuat program aplikasi berbasis *Window*.
- Untuk membuat objek-objek pembantu program seperti kontrol *ActiveX*, *file Help*, aplikasi internet dan sebagainya.
- Menguji program (*debugging*) dan menghasilkan program akhir berakhiran EXE yang bersifat *Executable* atau dapat langsung dijalankan.

## 2.11 CodeVision AVR [11]

*CodeVision-AVR* pada dasarnya merupakan perangkat lunak pemrograman mikontroler keluarga AVR berbasis bahasa C. Ada tiga komponen penting yang telah diintegrasikan dalam perangkat lunak ini: *Compiler C*, IDE dan Program generator.

*CodeVision-AVR* juga bisa digunakan untuk mengcompile sintaks c++ dan menghasilkan menjadi sebuah file .hex, dimana file .hex tersebut bisa dimasukkan ke dalam sebuah mikrokontroler yang kosong, sehingga mikrokontroler tersebut bisa digunakan.



**Gambar 2.13** Tampilan Awal Code Vision AVR

*CodeVisionAVR* juga menyediakan fungsi-fungsi tambahan yang sangat bermanfaat dalam pemrograman antarmuka AVR dengan perangkat luar yang umum digunakan dalam aplikasi kontrol.

Beberapa fungsi *library* yang penting diantaranya adalah fungsi-fungsi untuk pengaksesan LCD, komunikasi I2C, IC RTC (*Real time Integrated Development Environment* (IDE) telah diadaptasikan pada chip AVR yaitu *InSystem Programmer software*, Selain menu-menu pilihan yang umum dijumpai pada setiap perangkat lunak berbasis Windows, *CodeVisionAVR* ini telah mengintegrasikan perangkat lunak downloader yang bersifat In System Programmer (ISP) yang dapat digunakan untuk mentransfer kode mesin hasil kompilasi ke dalam sistem memori mikrokontrolerAVR yang sedang diprogram.

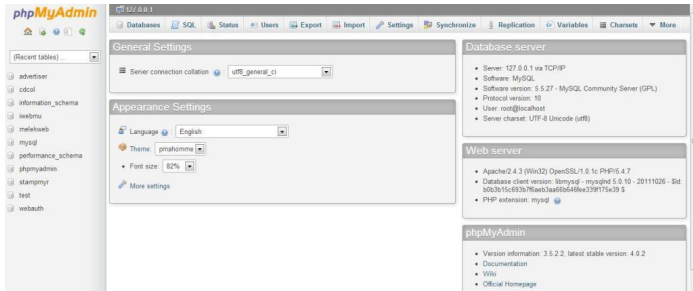
## 2.12 Database My SQL [12]

*Database* merupakan kumpulan file yang saling terintegrasi, namun *database* tidak akan dapat diakses oleh siapapun tanpa adanya software aplikasi. Aplikasi MySQL sebagai penyimpanan data sistem. MySQL merupakan sistem manajemen *database* yang digunakan untuk menyimpan data dalam Tabel terpisah [13]. MySQL merupakan *database server* yang paling sering digunakan dalam pemrograman PHP. MySQL digunakan untuk menyimpan data dalam *database* dan memanipulasi data-data yang diperlukan. Manipulasi data tersebut berupa menambah, mengubah, dan menghapus data yang berada dalam *database*.

Tahapan dalam membuat *database* MySQL Server adalah sebagai berikut : [14]

- *Install* aplikasi XAMPP pada *windows*.
- Kemudian masuk ke *phpmyadmin* di XAMPP. Ketik di *browser* alamat **localhost/phpmyadmin** sehingga muncul tampilan sebagai berikut.





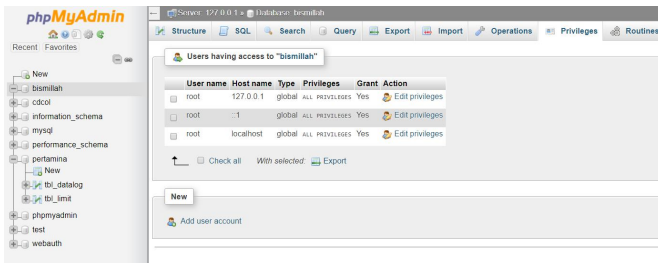
Gambar 2.14 Halaman Menu PhpMyAdmin

- Pada menu **Database**, silahkan masukkan nama *database* yang diinginkan, kemudian klik **Create**.



Gambar 2.15 Form Menu Database

- Selanjutnya menambahkan nama **user** untuk *database* yang baru saja kita buat. Silahkan masuk ke *database* yang sudah dibuat, Anda bisa menemukannya di bagian sebelah kiri. Begitu masuk, lihat menu **Privileges**. Kemudian klik **Add user**.



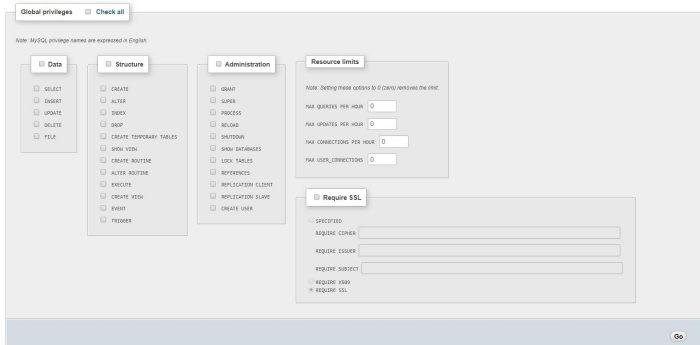
Gambar 2.16 Form Menu Privileges

- Setelah Anda klik **Add User** maka akan muncul form pembuatan user baru. Isikan **username**, Anda bisa mengisinya dengan nama apapun yang anda inginkan. Kemudian isikan juga **host**, gunakan **use text field** saja, lalu isikan dengan **localhost**. Selanjutnya isikan **password** 2 kali dan harus sama. Terakhir, ceklist **grant all privileges on database**.

The screenshot shows the 'Add user account' form in a MySQL management tool. The form is divided into two main sections: 'Login information' and 'Database for user account'. In the 'Login information' section, there are fields for 'User name', 'Host name', 'Password', 'Re-type', and 'Authentication Plugin'. The 'Host name' field is set to 'Any host' and '%'. The 'Authentication Plugin' is set to 'MySQL native password'. There is a 'Generate password' button and a 'Generate' button. In the 'Database for user account' section, there are three checkboxes: 'Create database with same name and grant all privileges', 'Grant all privileges on wildcard name (username)\_%', and 'Grant all privileges on database "bismillah"'. The third checkbox is checked.

**Gambar 2.17** Form Menu *Add User*

- Scroll ke bawah, dan Anda akan menemukan opsi **Global Privileges** guna memberikan hak akses kepada user untuk mengelola semua **database**, klik **Check All** . Setelah itu, klik tombol **Add User**.



**Gambar 2.18** Form Menu *Add User 2*

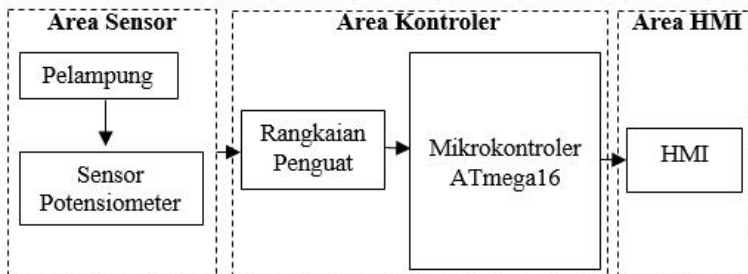
# BAB III

## PERANCANGAN *MONITORING* VOLUME LUMPUR

Pada bab ini akan dijelaskan tahapan-tahapan yang ada didalam perancangan *monitoring* volume lumpur ini. Perancangan tersebut dibagi menjadi beberapa bagian yaitu perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*), serta perancangan mekanik yang membantu kerja sistem. Perancangan pembuatan perangkat keras berupa penyusunan komponen- komponen elektronika menjadi satu kesatuan sistem rangkaian yang bisa bekerja sesuai yang diharapkan, sedangkan perancangan pembuatan perangkat lunak meliputi program yang dibutuhkan dalam sistem yang berhubungan dengan pemrograman. Sementara perancangan mekanik digunakan untuk mendukung jalannya alat tersebut.

### 3.1 Konfigurasi Sistem

Penelitian ini menjelaskan bagaimana sistem yang dibuat yang akan dijelaskan pada Gambar 3.1



**Gambar 3.1** Bagan Alur Kerja dari Alat

Pada Gambar 3.1 menjelaskan alur sistem *monitoring* terbagi menjadi 3 area, yaitu area sensor, area kontroler, dan area HMI. Mikrokontroler sebagai kontroler atau sebagai pusat pengontrol sistem.

Cara kerja dari alat ini, ketika catu daya dihidupkan, maka arus akan mengalir sehingga menyebabkan sistem tersebut aktif. Ketika sistem tersebut aktif, maka mikrokontroler mendapatkan masukan dari sensor *level* (potensiometer) dengan resistor yang nilainya berubah-ubah, perubahan putaran potensiometer disebabkan karena adanya pelampung bergerak naik turun sehingga menyebabkan potensiometer berputar sesuai dengan ketinggian lumpur. Sensor tersebut akan membaca data *analog* berupa nilai tegangan. Nilai tegangan tersebut masuk ke dalam ADC (*Analog Digital Converter*) mikrokontroler yang dikonversikan menjadi data digital. Kemudian data digital ketinggian yang telah terukur tersebut akan ditampilkan pada HMI.

Untuk mendapatkan alat sistem *monitoring* yang berjalan sesuai harapan diperlukan bagian pokok, yaitu:

- Sensor Potensiometer, merupakan sensor yang mendeteksi ketinggian dari lumpur berdasarkan perubahan dari nilai resistornya.
- Rangkaian Penguat, merupakan rangkaian yang digunakan untuk menentukan berapa penguatan yang digunakan sesuai dengan alat yang dibuat.
- Mikrokontroler ATmega16, mikrokontroler yang berfungsi sebagai pengendali, pengolah sinyal masuk dan keluar.
- HMI (*Human Machine Interface*), digunakan sebagai *monitoring* volume lumpur

Diagram blok pada Gambar 3.1 diatas menjelaskan mengenai alur kerja komponen – komponen yang sangat penting digunakan dalam membuat suatu alat atau sistem. Penjelasan dari masing-masing komponen dijelaskan pada sub bab perancangan perangkat keras mekanik, perancangan perangkat keras elektrik, dan perancangan perangkat lunak (*software*).

### 3.2 Perancangan Perangkat Keras Mekanik

Perancangan mekanik ini terdiri dari beberapa bagian yaitu perancangan mekanik sensor level, Perancangan mekanik sistem minimum, dan rancangan mekanik secara keseluruhan.

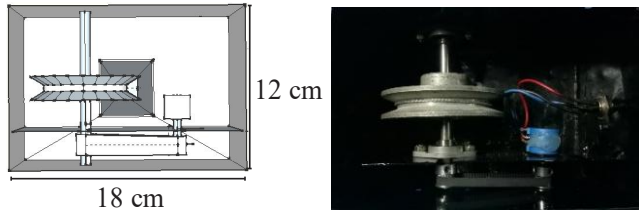
#### 3.2.1 Perancangan Mekanik Sensor *Level*

Perancangan packaging untuk mekanik sensor ini didesain untuk dapat mengukur ketinggian hingga mencapai 213,17 cm yang dibutuhkan pada industri pengeboran saat pengukuran *volume* lumpur yang terdapat pada *mudtank*. Perancangan mekanik sensor tersebut terbagi menjadi 2 bagian yaitu kotak sensor dan tiang pendiri sensor.

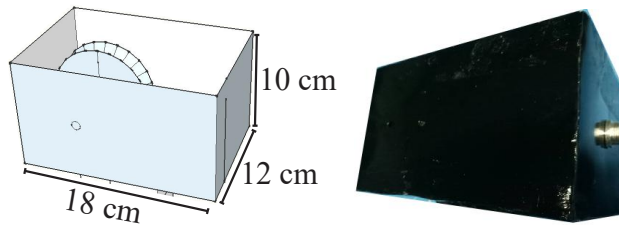
Pada perancangan kotak sensor terdapat beberapa komponen penyusunnya yaitu :

- Potensiometer berfungsi sebagai sensor pendeteksi perubahan dari naik turunnya cairan pada mud tank.
- *Teeth pulley* 8 mm berfungsi sebagai penggerak dari potensiometer dimana potensiometer tersebut juga akan memutar rangkaian *pulley*.
- *No teeth pulley* 3,5 inch berfungsi sebagai poros putar dari pergerakan rangkaian *pulley*.
- *Fixed Bearing* berfungsi sebagai pemutar rangkaian *pulley* dan sebagai penahan ujung-ujung as pejal.
- As pejal bulat berfungsi sebagai penghubung antara *pulley* 3,5 inch dan *teeth pulley* 8 mm. As pejal tersebut mempunyai spesifikasi panjang 10 cm dan diameter 8 mm.
- Kawat seling sepanjang 3 m berfungsi sebagai penghubung antara pelampung berbahan busa dengan pemberat yang terdapat didalam pipa *hollow*.
- Pelampung berbahan busa dengan ukuran panjang 10 cm, lebar 8 cm, dan tinggi 2 cm. Pelampung berfungsi sebagai media pendeteksi gerak dari ketinggian lumpur

Berikut merupakan perancangan kotak sensor bagian dalam terlihat pada Gambar 3.2 sedangkan bagian luar terlihat pada Gambar 3.3.



**Gambar 3.2** Perancangan Kotak Sensor Tampak Dalam



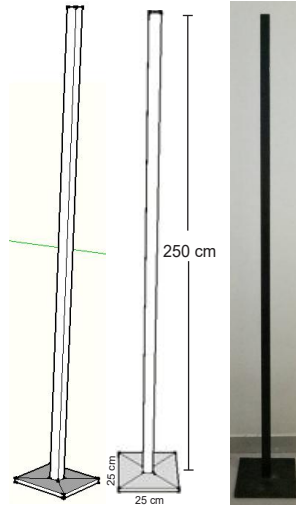
**Gambar 3.3** Perancangan Kotak Sensor Tampak Luar

Spesifikasi mekanik kotak sensor tersebut berukuran dimensi yaitu Panjang = 18 cm, Lebar = 12 cm, dan Tinggi = 10 cm. Didalam kotak sensor terdapat sistem rangkaian pulley yang terdiri dari pulley berukuran 3,5 inch, gear shaft berukuran 0,8 inch, dan belt shaft berukuran diameter 8 cm.

Sedangkan pada perancangan tiang pendiri sensor hanya terdapat 2 komponen yaitu :

- Lempengan besi berfungsi sebagai pemberat ketika dimasukkan dalam air atau lumpur dengan ukuran 25 cm x 25 cm x 1 cm dan mempunyai berat sebesar  $\pm 15$  kg.
- Pipa *hollow* 40 mm dengan tinggi 2,5 meter. Pipa ini berfungsi sebagai penghubung antara lempengan besi dengan kotak sensor tersebut.

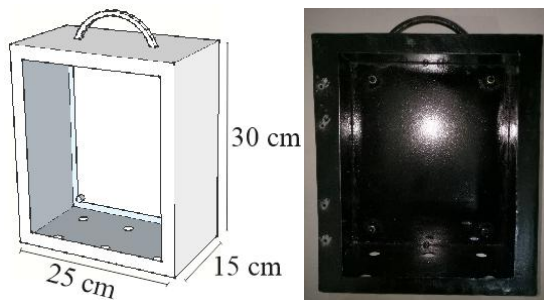
Berikut merupakan perancangan tiang pendiri sensor yang terlihat pada Gambar 3.4.



**Gambar 3.4** Perancangan Tiang Pendiri Sensor

### 3.2.2 Perancangan Mekanik Sistem Minimum

Perancangan mekanik sistem minimum ini seperti halnya *panel box* yang berbentuk balok. Perancangan ini berfungsi sebagai tempat menampung rangkaian sistem minimum ATmega16 dan rangkaian *Power supply*. Berikut perancangan mekanik sistem minimum yang terlihat pada Gambar 3.5. Spesifikasi dari mekanik ini mempunyai panjang 25 cm, lebar 15 cm, dan tinggi 30 cm. Dalam mekanik tersebut



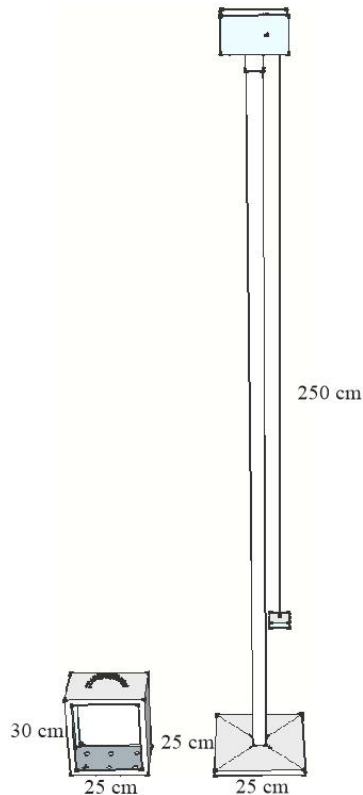
**Gambar 3.5** Perancangan Mekanik Sistem Minimum



terdapat lubang sebagai tempat diletakkan mikrokontroler, dibagian bawah kotak sensor terdapat 6 buah lubang yang digunakan sebagai penghubung sumber tegangan, penghubung antara sensor dengan mikrokontroler.

### 3.2.3 Perancangan Mekanik Keseluruhan

Perancangan mekanik secara keseluruhan dapat dilihat pada Gambar 3.6



**Gambar 3.6** Perancangan Mekanik Keseluruhan

Rancangan tersebut merupakan perancangan keseluruhan dari mekanik alat. Dari perancangan kotak sensor, tiang pendiri sensor dan

panel sistem minimum tersebut diintegrasikan menjadi satu sistem, dimana antara sensor dengan panel sistem minimum tersebut disambungkan dengan kabel, sehingga antara pecahan pecahan dari mekanik tersebut dapat terintegrasi dengan yang lainnya. Tujuan dari dirancangnya mekanik dari alat tersebut yaitu sebagai pendukung agar sistem yang dibuat bekerja dengan baik.

### **3.3 Perancangan Perangkat Keras Elektrik**

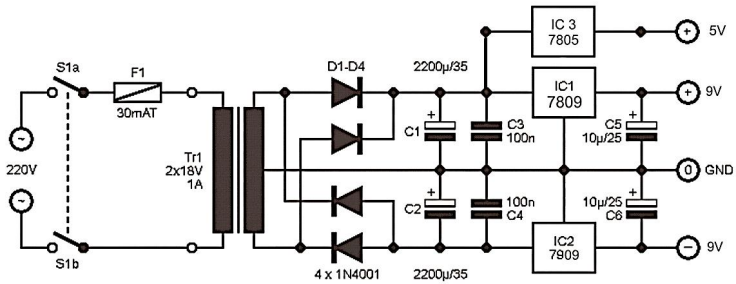
Perancangan perangkat keras elektrik ini terdiri dari lima bagian, yaitu perancangan *power supply*, perancangan rangkaian penguat, perancangan sistem minimum mikrokontroler ATmega16, perancangan komunikasi serial RS485 ke USB,

#### **3.3.1 Perancangan *Power supply*.**

*Power supply* merupakan suatu hardware komponen elektronika yang berfungsi sebagai pemasok tegangan arus listrik dengan terlebih dahulu merubah tegangan AC (*Alternating Current*) menjadi tegangan DC (*Direct Current*). *Power supply* ini digunakan sebagai sumber dari mikrokontroler ATmega16 yang digunakan saat sistem tersebut bekerja. Dan juga sebagai tegangan referensi untuk mengaktifkan penguatan dari rangkaian komponen IC LM741.

Alur kerja dari *power supply* ini adalah menyearahkan tegangan AC ke DC, namun tegangan 220 VAC diberi transformator *step down* dan bagian sekunder dari transformator tersebut dihubungkan dengan *diode bridge* sehingga hasilnya dari tegangan yang awalnya bolak-balik menjadi tegangan yang searah. Selanjutnya dihubungkan dengan kapasitor 2200uF/35V untuk menghaluskan gelombang yang tidak rata dari keluaran *diode bridge*.

Dari rangkaian *power supply* ini, dirancang untuk dapat mengeluarkan tegangan sebesar 5 volt, 9 volt dan -9 volt. Sehingga kami menggunakan IC Regulator dengan tipe LM7805, LM7809, dan LM7909. Fungsi dari ic ini hanya membatasi tegangan saja bukan untuk menambah tegangan keluaran. Rangkaian *power supply* dapat dilihat pada Gambar 3.7.



**Gambar 3.7** Skematik Regulator Tegangan 5v, 9v dan -9v

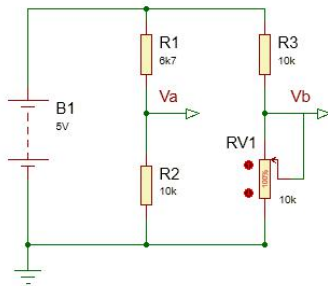
### 3.3.2 Perancangan Rangkaian Jembatan *Wheatstone* dan ASC

Rangkaian jembatan *wheatstone* merupakan sebuah rangkaian yang berfungsi sebagai mengukur suatu tahanan untuk memperoleh ketelitian dalam melakukan pengukuran tahanan yang nilainya relatif kecil sekali. Jembatan *wheatstone* adalah alat yang paling umum digunakan untuk pengukuran tahanan yang teliti dalam daerah 1 sampai 10.000  $\Omega$ . Perancangan kali ini, data tegangan yang diinginkan yaitu tegangan minimal sebesar 0 volt dan tegangan maksimal 2,5 volt. Untuk membuat kedua rangkaian ini dibutuhkan komponen yang sesuai agar hasil tegangan yang keluar dari kedua rangkaian ini sesuai dengan yang diinginkan. Berikut adalah data tegangan yang diinginkan dari rangkaian jembatan *wheatstone* dapat dilihat pada Tabel 3.1.

**Tabel 3.1** Data Jembatan *Wheatstone* yang Diinginkan

No.	Data tegangan <i>Wheatstone Bridge</i>	
1.	Tegangan Minimal	0 Volt
2.	Tegangan Maksimal	2,5 Volt

Gambar 3.8 merupakan rangkaian jembatan *wheatstone* dimana terdapat titik Va dan Vb untuk melihat beda potensial setelah adanya perubahan nilai RV1 (*sensor ten turn potentiometer*)



Gambar 3.8 Rangkaian *Wheatstone Bridge*

- Tegangan pada simulasi proteus  
 $V_A = 2.99$  Volts dan  $V_B = 2.5$  Volts.
- Perhitungan secara teori.

a. Ketika  $R_x$  bernilai maksimal (10k Ohm) maka :

$$V_A = \left( \frac{R_2}{R_1 + R_2} \right) \cdot V_{CC} \dots \dots \dots (3.1)$$

$$V_A = \left( \frac{10k}{6,7k + 10k} \right) \cdot 5 \text{ v} \dots \dots \dots (3.2)$$

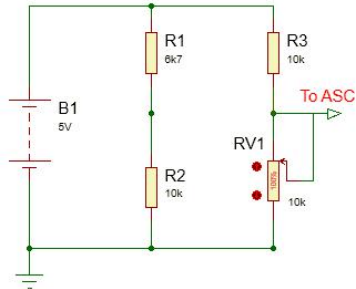
$$V_A = 3 \text{ v} \dots \dots \dots (3.3)$$

$$V_B = \left( \frac{R_x}{R_x + R_3} \right) \cdot V_{CC} \dots \dots \dots (3.4)$$

$$V_B = \left( \frac{10k}{10k + 10k} \right) \cdot 5 \text{ v} \dots \dots \dots (3.5)$$

$$V_B = 2,5 \text{ v} \dots \dots \dots (3.6)$$

Gambar 3.9 merupakan rangkaian *wheatstone bridge* dimana *output* yang dari rangkaian tersebut menuju ke rangkaian ASC. Rangkaian ASC merupakan suatu rangkaian konversi sinyal *analog* menjadi bentuk yang lebih sesuai yang merupakan antarmuka dengan elemen-elemen lain dalam suatu kontrol proses. Pada perancangan kali ini rangkaian tersebut digunakan sebagai penguat untuk menguatkan tegangan sesuai dengann tegangan yang diinginkan. Pada Gambar 3.10 merupakan rangkaian ASC, dimana rangkaian mendapat *input* dari

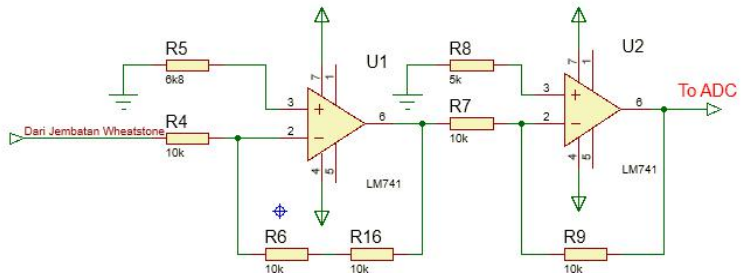


**Gambar 3.9** Rangkaian *Wheatstone Bridge to pin ASC*.

rangkaian jembatan *wheatstone*. Rangkaian ini digunakan untuk merubah tegangan dari *input* jembatan *wheatstone* dari 0 Volt sampai 2,5 Volt menjadi 0 Volt sampai 5 Volt.

**Tabel 3.2** Data Tegangan ASC yang Diinginkan

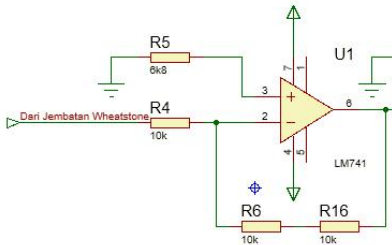
No.	Data tegangan ASC yang diinginkan	
1.	Tegangan Minimal	0 Volt
2.	Tegangan Maksimal	5 Volt



**Gambar 3.10** Rangkaian *Analog Signal Conditioning*

- Tegangan pada simulasi proteus  
Vout = 5 Volt

- Perhitungan secara teori.



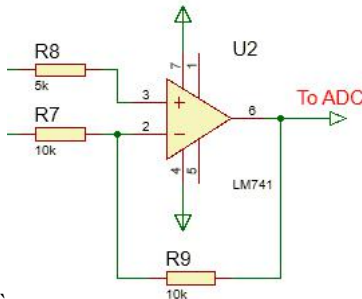
Diketahui :  
 $R_f = 20k \Omega$   $R_i = 10k$   
 $\Omega$   $V_{in} = 2,5$  volt

**Gambar 3.11** Rangkaian Op-Amp 1

$$A_v = -\frac{R_f}{R_i} \cdot V_{in} \dots \dots \dots (3.7)$$

$$A_v = -\frac{5k}{2k5} \cdot 2,5 \text{ v} \dots \dots \dots (3.8)$$

$$A_v = -5 \text{ v} \dots \dots \dots (3.9)$$



Diketahui :  
 $R_f = 10k \Omega$   $R_i = 10k$   
 $\Omega$   $V_{in} = -5$  volt

**Gambar 3.12** Rangkaian Op-Amp 2

$$A_v = -\frac{R_f}{R_i} \cdot V_{in} \dots \dots \dots (3.10)$$

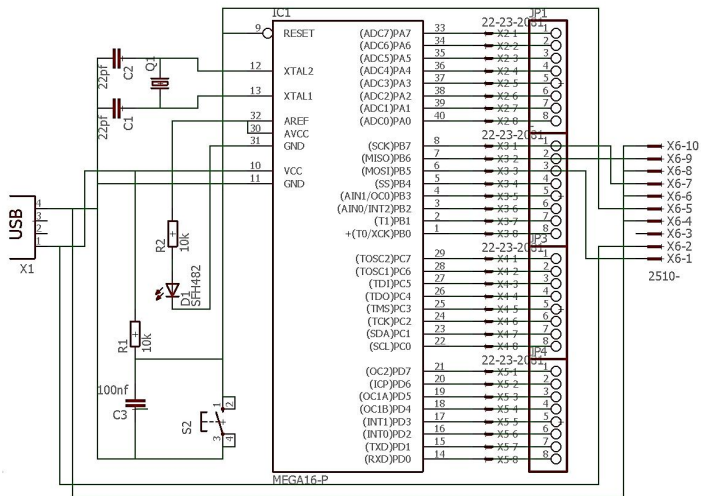
$$A_v = -\frac{10k}{10k} \cdot -5 \text{ v} \dots \dots \dots (3.11)$$

$$A_v = 5 \text{ v} \dots \dots \dots (3.12)$$

Sehingga, tegangan keluaran dari rangkaian tersebut sebesar 5 volt.

### 3.3.3 Perancangan Sistem Minimum Mikrokontroler ATmega16

Perancangan sistem minimum mikrokontroler ini merupakan suatu rangkaian elektronik minimum yang diperlukan untuk beroperasinya IC mikrokontroler. Pada umumnya, suatu mikrokontroler membutuhkan dua elemen (selain *power supply*) untuk berfungsi yaitu Kristal Oscillator dan Rangkaian *Reset*. Sisteem minimum ini kemudian bisa dihubungkan dengan rangkaian lain untuk menjalankan fungsi tertentu. Penelitian ini menggunakan rangkaian sistem minimum ATmega16 yang berfungsi untuk pengendali dari rangkaian ini. ATmega16 merupakan mikrokontroler yang sangat mudah dicari dan banyak digunakan oleh para programmer. Dengan fitur-fitur yang dimiliki ATmega16 dan jumlah memori yang dimiliki ATmega16 maka tidak ada hal yang perlu diragukan kembali untuk mempergunakannya. Kristal dan dua buah kapasitor sebagai osilator rangkaian sistem minimum yang berfungsi untuk memenuhi kebutuhan clock pada ATmega16. Berikut merupakan Gambar Rangkaian sistem minimum ATmega16 terlihat pada Gambar 3.13.



Gambar 3.13 Rangkaian Sistem Minimum ATmega16

Dari Gambar 3.13 dapat dilihat terdapat beberapa komponen pada mikrokontroler AVR ATmega16 yang digunakan dalam sistem minimum ini, antara lain:

- *Sistem reset.*

Sistem *reset* berfungsi untuk *mereset* mikrokontroler jika ingin mengulang proses yang sedang dijalankan mikrokontroler dari awal. Mikrokontroler AVR ATmega16 memiliki sistem *reset* internal pada pin 9 (*RESET*). Sistem *reset* menggunakan resistor 10 K $\Omega$  dan kapasitor 100 nF. Untuk *mereset* digunakan sebuah *push button*.

- *Oscillator Crystal* (XTAL1 dan XTAL2)

Pada sistem minimum ini digunakan *Oscillator Crystal* 12 MHz sebagai detak bagi mikrokontroler dengan komponen tambahan berupa dua buah kapasitor keramik 22 pF. Komponen ini bertujuan untuk mempercepat pemrosesan data atau menghasilkan sinyal dengan tingkat kestabilan frekuensi yang sangat tinggi.

- *Pin Input / Output*

*Pin Input / Output* adalah pin terdapat dalam ATmega16 yang dapat digunakan sebagai *port* masukan (*input*) atau *port* keluaran (*output*). ATmega16 memiliki 4 buah port yaitu *port A*, *port B*, *port C*, dan *port D*. *Port* yang dapat dijadikan *port* masukan atau *port* keluaran adalah *port B*, *port C*, dan *port D*.

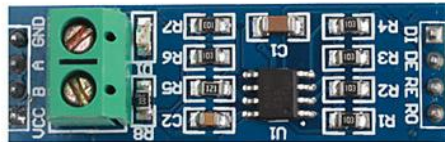
### 3.3.4 Perancangan Komunikasi Serial RS485 ke USB

Dalam perancangan komunikasi serial menggunakan modul serial RS485 ke USB. Modul ini merupakan modul serial yang mengkonversi RS485 ke USB begitu juga sebaliknya. Rangkaian ini memanfaatkan antarmuka UART dengan cara menghubungkan pin TX dan RX yang berfungsi sebagai pengirim dan penerima data terhubung dengan pin RX dan TX pada IC RS485. Berikut merupakan Gambar dari modul komunikasi RS485 ke USB.

Rangkaian komunikasi serial merupakan suatu rangkaian yang dapat berfungsi untuk menghubungkan antara mikrokontroler dengan



komputer (PC), sehingga data yang dihasilkan dapat ditampilkan dan disimpan pada komputer. Rangkaian komunikasi yang digunakan pada perancangan ini menggunakan modul rangkaian RS485, dimana komunikasi serial ini dapat menjangkau jarak yang lebih jauh sekitar  $\pm 1200$  meter.



**Gambar 3.14** Modul Rangkaian RS485

### 3.4 Perancangan Perangkat Lunak

Pada perancangan perangkat lunak ini lebih memfokuskan pada bagaimana cara kerja dari sistem yang dibuat, biasanya disebut dengan nama *flowchart*. *Flowchart* merupakan suatu metode yang digunakan untuk memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Bagan ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung. *Flowchart* ini merupakan langkah awal pembuatan program. Dengan adanya *flowchart* urutan proses kegiatan menjadi lebih jelas.

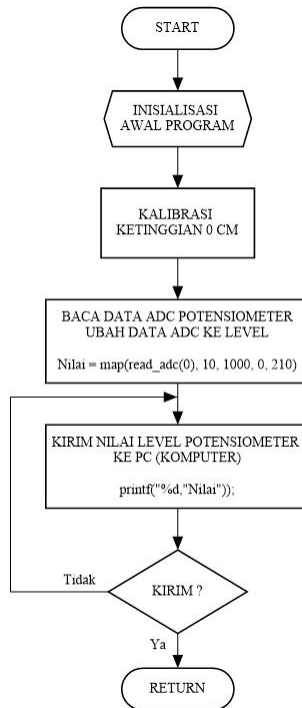
Untuk pengolahan data dengan komputer, dapat dirangkum urutan dasar untuk pemecahan suatu masalah, yaitu;

- *START*: berisi instruksi untuk persiapan peralatan yang diperlukan sebelum menangani pemecahan masalah.
- *READ*: berisi instruksi untuk membaca data dari suatu peralatan.
- *PROCESS*: berisi kegiatan yang berkaitan dengan pemecahan persoalan sesuai dengan data yang dibaca.
- *WRITE*: berisi instruksi untuk merekam hasil kegiatan ke peralatan *output*.
- *END*: mengakhiri kegiatan pengolahan [3].

Dalam perancangan perangkat lunak ini akan membahas tentang algoritma pembacaan sensor ketinggian dan komunikasi serial RS485, program nilai pembacaan sensor ketinggian, perancangan HMI (*Human Machine Interface*) dan program pembuatan rancangan visual pada HMI.

### 3.4.1 Algoritma Pembacaan Sensor Ketinggian dan Komunikasi RS485.

Dalam pembacaan nilai dari sensor ketinggian ini didapatkan nilai data yang telah di proses oleh mikrokontroler. Algoritma pembacaan sensor ini difokuskan pada pembacaan sensor ketinggian dalam mikrokontroler, yaitu bagaimana cara agar bisa mendapatkan



Gambar 3.15 Flowchart Mikrokontroler

nilai ketinggian yang sesuai dengan keinginan atau sesuai dengan kebutuhan. Kemudian nilai pembacaan sensor tersebut dikirimkan ke *personal komputer* (PC) melalui komunikasi serial RS485 menggunakan sistem satu arah (*halfduplex*), sehingga data yang telah diproses dari mikrokontroler tersebut dapat terlihat pada PC.

### 3.4.2 Program Pembacaan Sensor Ketinggian

```
while (1)
{
    printf("%d,",map(read_adc(0), 10, 1000, 0, 210));
    printf("%d,",map(read_adc(1), 10, 1000, 0, 210));
    printf("%d",map(read_adc(2), 10, 1000, 0, 210));
    printf("\n\r");
    delay_ms(500);
}
```

**Gambar 3.16** Program Mapping Sensor

Program ini merupakan program untuk mendeteksi nilai ADC dan level. Kemudian dari kedua data tersebut nilai keluaran dari adc akan diubah atau diatur dari batas yang telah dicapai programnya sesuai dengan kebutuhan.

### 3.4.3 Perancangan *Human Machine Interface* (HMI)

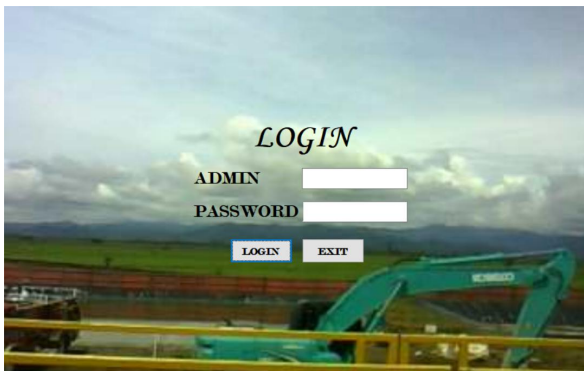
Perancangan HMI merupakan suatu komponen yang sangat penting dalam hal pengamatan atau *monitoring* dari setiap kondisi yang akan diamati dikarenakan HMI dapat dijadikan sebagai visualisasi dari kondisi *real* yang telah dibuat sesuai dengan keadaan dilapangan atau secara real. Dalam perancangan HMI ini terbagi menjadi 2 yaitu Algoritma HMI dan Perancangan *Visual* HMI.

#### 1.4.3.1 Perancangan *Visual Monitoring* HMI

Perancangan visual HMI merupakan suatu perangkat lunak yang dapat membuat para penggunannya tidak mengalami kebingungan saat ingin melakukan pengamatan pada objek yang diukur. Perancangan *visual* HMI terbagi menjadi 4 form, yaitu *form login*, *form channel*, *form grafik*, dan *form database*.

a. *Form Login*

Pada Gambar 3.17 merupakan rancangan visual HMI pada *form login*. *Form Login* ini terdapat 2 *text box* yaitu *admin* yang digunakan untuk menyetikkan *username* sedangkan *password* digunakan untuk menuliskan *password* yang telah diatur pada program. Setelah menyetikkan *username* dan *password* maka klik tombol *login* yang berfungsi untuk masuk ke dalam *channel monitoring*, sedangkan tombol *exit* digunakan untuk memberhentikan aplikasi *monitoring*.

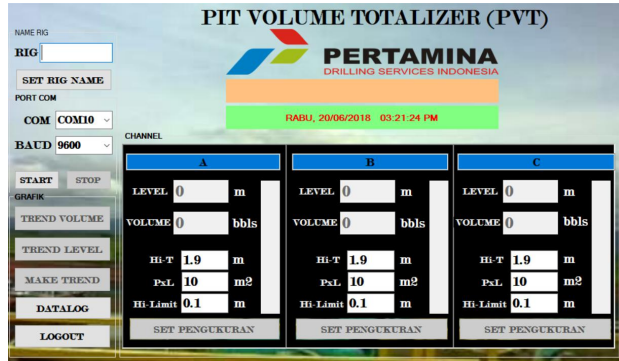


**Gambar 3.17** Tampilan Form Login

b. *Form Channel*

Pada Gambar 3.18 merupakan rancangan *visual HMI* pada *form channel* atau *form monitoring*. *Form channel* ini terdapat 4 bagian yang terbagi yaitu *Name Rig*, *Port COM*, *Grafik*, dan *Channel*. Dari 4 bagian tersebut mempunyai fungsi yang berbeda-beda. *Name rig* berfungsi untuk menuliskan nama rig atau nama tower untuk pengeboran di perusahaan PT. Pertamina Drilling Services Indonesia (PT. PDSI). *Port Com* digunakan untuk menyambungkan *port* mikrokontroler ke PC untuk membaca nilai serial yang telah diproses pada mikrokontroler. *Grafik* digunakan untuk menampilkan grafik level, volume, rata-rata level dan volume, dan datalog (penyimpanan *database*). Sedangkan *channel* digunakan sebagai *monitoring*

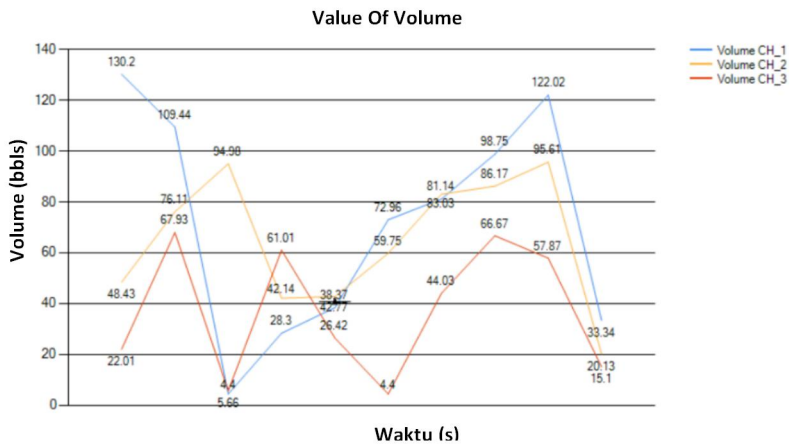
ketinggian cairan secara *real time*, *channel* ini terbagi menjadi 3 *channel* untuk mengukur di 3 tempat yang berbeda.



Gambar 3.18 Tampilan Form Menu Channel

c. *Form* Grafik

Pada Gambar 3.19 merupakan rancangan HMI untuk melihat secara *real time* grafik yang didapatkan dari proses pengukuran ketinggian volume cairan. Pada *Form* Grafik ini terdapat *form* volume



Gambar 3.19 Tampilan Form Grafik

yang digunakan untuk menampilkan grafik keadaan volume cairan secara realtime. *Form Level* digunakan untuk menampilkan grafik keadaan level cairan secara realtime. Sedangkan *Make Trend* digunakan untuk menampilkan grafik rata-rata pengukuran selama berjalannya *monitoring* keadaan lumpur tersebut.

d. Form *Database*

Pada Gambar 3.20 merupakan rancangan HMI untuk penyimpanan data yang telah diproses yang selanjutnya akan disimpan dengan format excel yang digunakan sebagai dokumentasi atau riwayat pemakaian dari alat *monitoring* volume lumpur tersebut. Dalam penyimpanan ini terbagi menjadi 2 penyimpanan, yaitu penyimpanan data secara *realtime* tiap 1 detik disimpan, dan penyimpanan ketika dalam keadaan penuh dan habis.

DATA SENSOR					ALARM RECORD					
No	Date	Time	Volume	Level	No	Date	Time	PIT	Value	SP
1	25/05/2018	01:06:15 PM	4.40	0.07	1	25/05/2018	01:14:39 PM	A	1.9	1.9
2	25/05/2018	01:06:16 PM	71.70	1.14	2	25/05/2018	01:14:42 PM	A	1.93	1.9
3	25/05/2018	01:06:17 PM	26.42	0.42	3	25/05/2018	01:14:45 PM	A	2.06	1.9
4	25/05/2018	01:06:18 PM	81.77	1.3	4	25/05/2018	01:14:47 PM	A	0.04	0.1
5	25/05/2018	01:06:19 PM	119.51	1.9	5	25/05/2018	01:14:49 PM	A	1.9	1.9
6	25/05/2018	01:06:20 PM	42.14	0.67	6	25/05/2018	01:14:49 PM	A	0.09	0.1
7	25/05/2018	01:06:21 PM	30.82	0.49	7	25/05/2018	01:14:50 PM	A	0.1	0.1
8	25/05/2018	01:06:22 PM	83.65	1.33	8	25/05/2018	01:15:05 PM	A	1.94	1.9
9	25/05/2018	01:06:23 PM	83.65	1.33	9	25/05/2018	01:15:05 PM	A	2.02	1.9
10	25/05/2018	01:06:24 PM	108.81	1.73	10	25/05/2018	01:15:14 PM	A	0.03	0.1
11	25/05/2018	01:06:25 PM	35.22	0.56	11	25/05/2018	01:15:18 PM	A	2.07	1.9
12	25/05/2018	01:06:26 PM	103.15	1.64	12	25/05/2018	01:15:19 PM	A	0.05	0.1
13	25/05/2018	01:06:27 PM	45.92	0.73	13	25/05/2018	01:15:21 PM	A	2.03	1.9
14	25/05/2018	01:06:28 PM	40.25	0.64	14	25/05/2018	01:15:22 PM	A	0.02	0.1
15	25/05/2018	01:06:29 PM	59.12	0.94	15	25/05/2018	01:15:26 PM	A	0.04	0.1
16	25/05/2018	01:06:30 PM	103.15	1.64	16	25/05/2018	01:15:33 PM	A	0.06	0.1
17	25/05/2018	01:06:31 PM	89.32	1.42	17	25/05/2018	01:15:36 PM	A	0.04	0.1
18	25/05/2018	01:06:32 PM	15.10	0.24	18	25/05/2018	01:15:58 PM	C	1.96	1.9
19	25/05/2018	01:06:33 PM	61.64	0.98	19	25/05/2018	01:15:59 PM	B	1.99	1.9
20	25/05/2018	01:06:34 PM	120.14	1.81	20	25/05/2018	01:15:59 PM	C	0	0.1

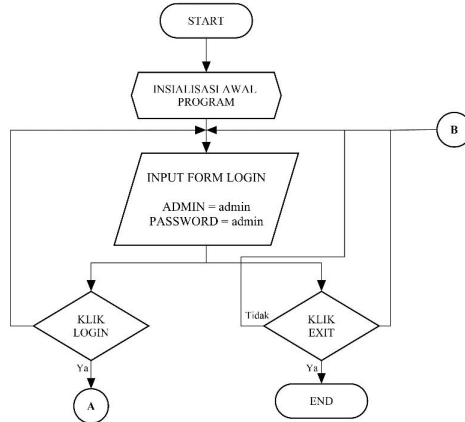
Gambar 3.20 Tampilan Form *Database*

1.4.3.2 Flowchart HMI

Berikut merupakan algoritma dari visual HMI yang telah dibuat untuk mempermudah cara penggunaan HMI.

Pada Gambar 3.21 – 3.25 merupakan algoritma bagaimana cara penggunaan dari HMI (*Human Machine Interface*) dari awal hingga akhir. Flowchart tersebut terbagi menjadi 5 bagian yaitu *form login*, *form channel*, *form grafik*, *form logout* dan *form database*.

a. *Form Login*



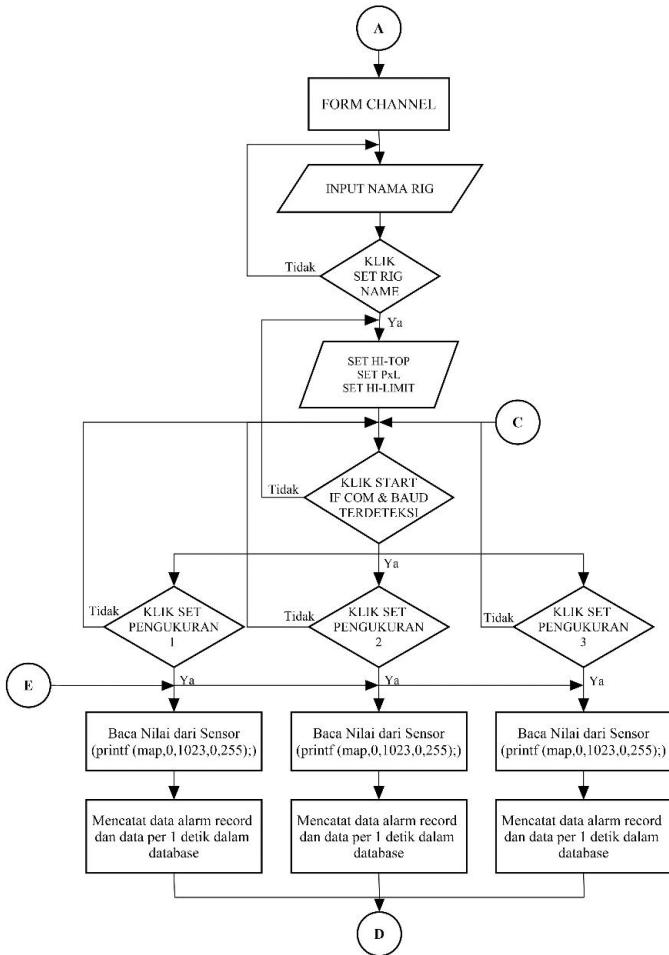
**Gambar 3.21** *Flowchart Form Login*

Sebelum mengakses tampilan utama dalam *monitoring*, pengguna diharuskan untuk mempunyai akun yang digunakan untuk *login*. Jika pengguna tidak mempunyai akun yang benar maka orang tersebut tidak dapat mengakses atau menjalankan aplikasi tersebut. HMI hanya digunakan oleh operator tertentu untuk dapat menjalankan *interface tersebut*. Saat tampilan awal muncul maka mengisi *admin* dan *password* lalu klik *login*. Jika saat memasukkan akun tersebut benar maka akan menuju ke tampilan *monitoring*, sedangkan jika salah maka akan mencoba kembali login pada tampilan *login*.

b. *Form Channel*

Pada tampilan ini digunakan untuk mengawasi volume yang terdapat pada mudtank. Setelah sukses masuk ke dalam tampilan *monitoring*, maka mengatur nama rig, lalu mengatur batas tinggi penuh

dan batas tinggi habis, serta mengatur luas media yang dipakai. Sambungkan koneksi serial dengan menentukan COM dan BAUDRATE, kemudian klik set pengukuran 1,2 atau 3 yang akan



**Gambar 3.22** Flowchart Form Channel

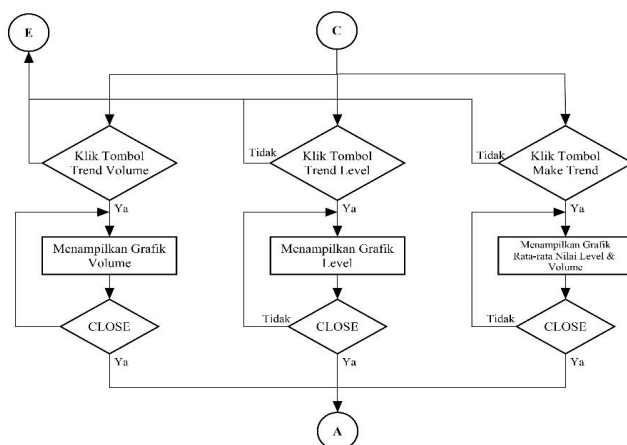
diukur, maka pembacaan data ketinggian dan data volume telah dapat dilakukan, lalu data tersebut disimpan dalam database yang telah



disediakan. Penyimpanan tersebut terbagi menjadi 2 yaitu penyimpanan saat keadaan penuh atau habis (alarm record) dan penyimpanan data per 1 detik pengukuran.

c. *Form Grafik*

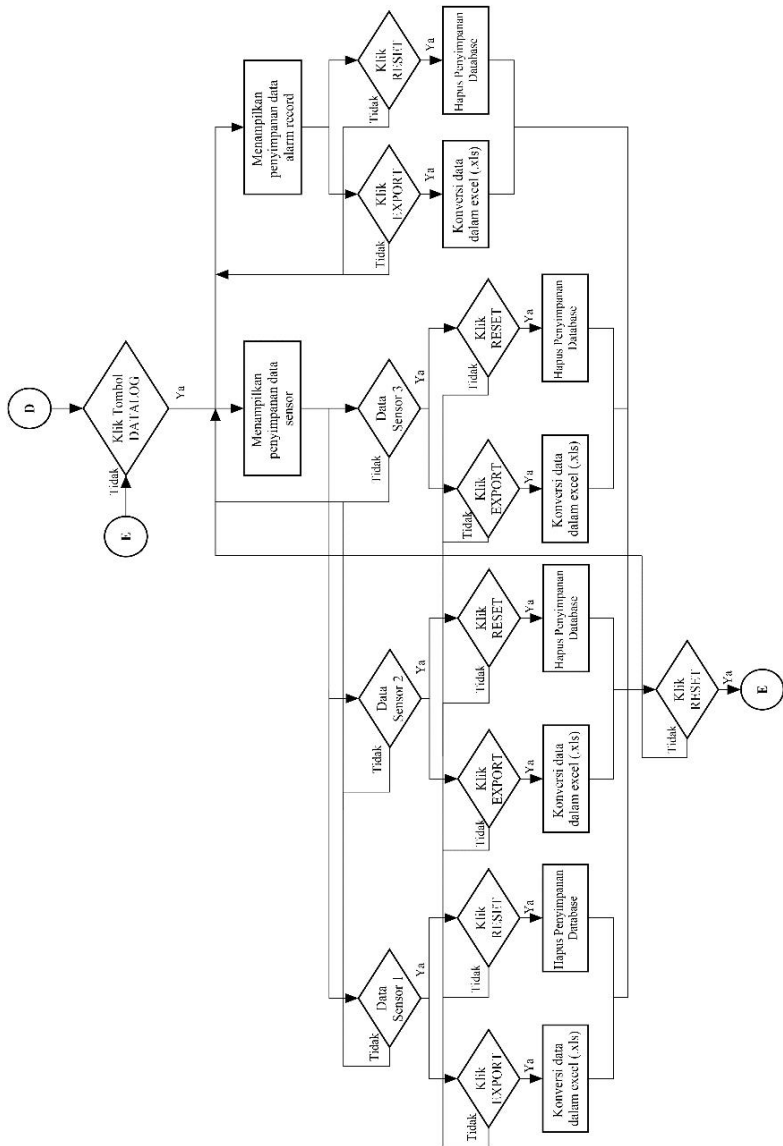
Flowchart tampilan grafik ini digunakan untuk melihat grafik keadaan *level*, volume, serta rata-rata grafik *level* dan volume. Ketika sudah membaca data yang telah ditampilkan maka pilih grafik yang ingin dilihat pada *monitoring*.



**Gambar 3.23** *Flowchart Form Grafik*

d. *Form Database*

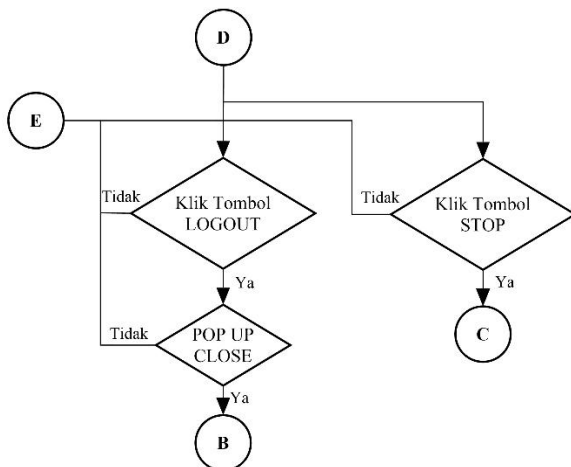
Dalam penyimpanan data ketinggian dan volume pada *monitoring* menggunakan *database MySQL*. Ketika ingin menyimpan data yang dibutuhkan untuk pengawasan maka klik tombol "DATALOG", lalu akan menampilkan data pada penyimpanan per 1 detik dan penyimpanan hanya saat keadaan penuh dan habis. Tiap penyimpanan tersebut dapat menyimpan data tiap channel dengan memilih channel berapa yang akan disimpan, lalu pilih *export (save to .xls)* atau *reset (menghapus data)*, *refresh (mengambil data yang baru)*.



Gambar 3.24 Flowchart Form Database

e. *Form Logout*

Ketika ingin mengeluarkan tampilan *monitoring*, maka Klik tombol “LOGOUT” pada tampilan tersebut, Jika iya maka akan muncul “POP UP CLOSE” lalu klik “YA”. Jika ingin memberhentikan pembacaan serial maka klik tombol “STOP”, sehingga pembacaan tersebut akan berhenti.



**Gambar 3.25** *Flowchart Form Logout*

## **BAB IV**

### **HASIL PENGUJIAN SISTEM *MONITORING***

### **VOLUME LUMPUR**

Pada bab ini akan dibahas mengenai pengujian dari sistem alat, pengujian sistem perangkat keras elektrik, pengujian perangkat lunak serta pengujian *Human Machine Interface* (HMI). Tujuan dilakukannya pengujian pada sistem ini untuk mengetahui keberhasilan dari keseluruhan sistem dan untuk mengetahui kesesuaian antara teori dengan hasil perancangan. Pengujian dilakukan pada masing-masing bagian terlebih dahulu, kemudian dari masing-masing bagian tersebut diuji secara keseluruhan. Setelah proses pengujian, langkah selanjutnya adalah menganalisa dari hasil pengujian tersebut.

#### **4.1 Pengujian Perangkat Keras Elektrik**

Perangkat keras elektrik ini merupakan komponen penyusun rangkaian yang penting dari rancangan sistem. Pengujian perangkat keras elektrik ini bertujuan untuk mengetahui keberhasilan dari tiap rangkaian elektronika yang telah dirancang untuk mendukung sistem yang dibuat. Pada pengujian ini meliputi beberapa pengujian terhadap komponen-komponen yang digunakan, yaitu pengujian rangkaian *power supply*, pengujian sensor potensiometer, pengujian rangkaian penguat, pengujian sistem minimum ATmega16, pengujian rangkaian *modul converter* RS485.

##### **4.1.1 Pengujian Rangkaian *Power supply***

Rangkaian *power supply* adalah rangkaian yang paling penting untuk dilakukan pengujian terlebih dahulu dikarenakan rangkaian ini berfungsi sebagai pemasok tegangan yang diperlukan oleh sebuah mikrokontroler untuk memproses suatu sistem.

Pada perancangan yang dibuat tegangan yang dibutuhkan dalam sistem ini yaitu sebesar 5 volt, 9 volt, dan -9 volt. Sehingga menggunakan 3 jenis IC regulator untuk dapat menstabilkan tegangan

yang diinginkan. Pengujian ini terbagi menjadi 3 pengujian yaitu pengujian *power supply* dengan tegangan 5 volt, tegangan 9 volt, dan tegangan -9 volt. Dalam pengujian tersebut kami menggunakan cara dengan mengukur tegangan keluaran dari rangkaian dengan menggunakan multimeter digital dimana *probe* positif kami tempelkan pada tegangan positif, dan *probe* negatif kami tempelkan di bagian ground setelah memperoleh data untuk pengukuran pertama, maka pengukuran kedua yaitu dengan mengganti dengan stop kontak yang berbeda. Berikut merupakan hasil pengujian tegangan *power supply* dengan tegangan 5 volt, 9 volt, dan -9 volt.

**Tabel 4.1** Pengujian Tegangan Regulator 7805 (5 volt)

Pengukuran ke-n	Tegangan Sumber AC (v)	Tegangan <i>Input</i> (v)	Tegangan <i>Output</i> (volt)
1.	220	15	4,93
2.	220	15	4,94
3.	220	15	4,93

**Tabel 4.2** Pengujian Tegangan Regulator 7809 (9 volt)

Pengukuran ke-n	Tegangan Sumber AC (v)	Tegangan <i>Input</i> (v)	Tegangan <i>Output</i> (volt)
1.	220	15	9,14
2.	220	15	9,13
3.	220	15	9,13

**Tabel 4.3** Pengujian Tegangan Regulator 7909 (-9 volt)

Pengukuran ke-n	Tegangan Sumber AC (v)	Tegangan <i>Input</i> (v)	Tegangan <i>Output</i> (volt)
1.	220	15	-9,24
2.	220	15	-9,25
3.	220	15	-9,24

Berdasarkan hasil pengujian pada Tabel 4.1 , dapat dilihat bahwa tegangan regulator LM7805 akan mendekati 5 volt bila tegangan masukannya diatas  $\pm 7$  volt, karena regulator LM7805 memiliki spesifikasi tegangan *input* minimal sebesar +7 volt pada datasheet. Tegangan keluaran regulator LM7805 yang ideal adalah 5 volt dengan toleransi  $\pm 4\%$ , namun berdasarkan Tabel diatas, rangkaian *power supply* ini memiliki keluaran tegangan yaitu sebesar 4,93 volt. Maka pada tegangan 5 volt ini masih terdapat dalam jangkauan range tegangan *output*. Tegangan 5 volt ini digunakan untuk tegangan *input* dari mikrokontroler ATmega16 yang dapat bekerja sesuai dengan spesifikasi tegangan yaitu sebesar 4,5 – 5,5 volt.

Berdasarkan hasil pengujian pada Tabel 4.2 , dapat dilihat bahwa tegangan regulator LM7809 akan mendekati 9 volt bila tegangan masukannya diatas +12 volt, karena regulator LM7809 memiliki spesifikasi tegangan *input* minimal sebesar +12 volt pada datasheet. Tegangan keluaran regulator LM7809 yang ideal adalah 9 volt dengan toleransi  $\pm 4\%$ , namun berdasarkan Tabel diatas, rangkaian *power supply* ini memiliki keluaran tegangan yaitu sebesar 9,14 volt. Maka pada tegangan 9 volt ini masih terdapat dalam jangkauan range tegangan *output*. Tegangan 9 volt ini digunakan untuk tegangan *input* referensi dari komponen IC LM741 yang berfungsi sebagai pengondisian sinyal *analog*.

Berdasarkan hasil pengujian pada Tabel 4.3 , dapat dilihat bahwa tegangan regulator LM7909 akan mendekati -9 volt bila tegangan masukannya diatas  $\pm 12$  volt, karena regulator LM7909 memiliki spesifikasi tegangan *input* minimal sebesar  $\pm 12$  volt pada *datasheet*. Tegangan keluaran regulator LM7909 yang ideal adalah -9 volt dengan toleransi  $\pm 4\%$ , namun berdasarkan Tabel diatas, rangkaian *power supply* ini memiliki keluaran tegangan yaitu sebesar -9,24 volt. Maka pada tegangan -9 volt ini masih terdapat dalam jangkauan *range* tegangan *output*. Tegangan -9 volt ini digunakan untuk tegangan *input* referensi dari komponen IC LM741 yang berfungsi sebagai pengondisian sinyal *analog*.

#### 4.1.2 Pengujian Sensor Level Potensiometer

Potensiometer adalah komponen resistor variabel yang dimana komponen tersebut nilai tahanannya atau resistansinya dapat diubah-ubah. Penelitian ini menggunakan potensiometer 10 kali putaran sebagai sensor dari ketinggian permukaan lumpur.

Pengujian sensor potensiometer ini bertujuan untuk mengetahui dan membandingkan nilai resistansi yang terdapat pada datasheet dengan nilai resistansi yang diukur langsung serta juga untuk memeriksa dari kondisi potensiometer tersebut dapat berfungsi dengan baik atau tidak. Pengujian ini juga bertujuan untuk mengetahui berapa besar perbandingan antara pengukuran secara manual dengan pengukuran secara otomatis oleh suatu sistem. Pengujian ini dilakukan pengukuran dengan menggunakan penggaris atau meteran.



**Gambar 4.1** Pengukuran Sensor Level Potensiometer

Pada Gambar 4.1 dapat terlihat saat pengukuran sensor level secara nyata. Potensiometer yang digunakan yaitu sebesar 10.000 ohm. Potensiometer yang digunakan memiliki toleransi sebesar  $\pm 5\%$  dari 10.000 ohm (9500-10500 ohm) sehingga dapat dikatakan bahwa potensiometer masih dapat digunakan. Ketika dilakukan pengukuran pada setiap putaran, maka didapatkan hasil bahwa dalam satu kali putaran potensiometer dapat mencapai panjang  $\pm 21$  cm, karena ada 10 kali putaran, maka ketinggian maksimal yang dapat terukur potensiometer ini adalah  $\pm 210$  cm atau  $\pm 2,1$  meter. Sebelum dilakukan

pengukuran, maka harus dipastikan bahwa alat telah terkalibrasi dan dapat mengukur mulai dari 0, oleh karena itu dilakukanlah pengujian pengukuran untuk menentukan kalibrasi alat. Setelah mendapatkan nilai pengukuran secara manual yaitu 210 cm, maka dilakukan pengukuran pertama dengan rangkaian untuk mengukur berapa panjang yang terukur setiap 1 kali putaran (dari satu titik kembali ke titik yang sama) dengan memasukkan nilai maksimal 210. Berikut ini Tabel 4.4 menerangkan hasil pengukuran setiap putaran.

**Tabel 4.4** Pengujian Sensor Level Potensiometer

Putaran ke-n	Hasil terukur (m)	Selisih dengan putaran sebelumnya (m)	Tegangan terukur pada kaki data (v)	Hambatan yang terukur ( $\Omega$ )	Tegangan terukur pada pin ADC (v)
0	0	0	0	0,6	0.0001
1	0,21	0,21	0,253	1046	0,515
2	0,43	0,22	0,505	2153	1,023
3	0,63	0,20	0,761	3021	1,531
4	0,84	0,21	1,032	4125	2,065
5	1,06	0,22	1,235	5053	2,480
6	1,26	0,20	1,516	6051	3,032
7	1,47	0,21	1,734	7110	3,468
8	1,68	0,21	2,025	8143	4,051
9	1,88	0,20	2,251	9081	4,503
10	2,1	0,22	2,415	10010	4,832
Total ukur	2,1	-	2,415	10010	4,832

Pada pengujian sensor level potensimeter ini menggunakan tegangan referensi  $\pm 5$  volt. Berdasarkan data yang telah diambil dapat dilihat pada Tabel 4.4 ketika potensiometer diputar sebanyak 10 kali putaran maka hambatan yang terukur yaitu sebesar 10.010  $\Omega$ . Pada saat



putaran ke 0 nilai hambatan yaitu sebesar  $10 \Omega$  sedangkan pada hasil yang terukur yaitu sebesar 0 cm dengan selisih putaran sama dengan hasil yang terukur (cm), dan tegangan yang terukur pada kaki data yaitu sebesar 0 volt. Perbedaan antara putaran selanjutnya dengan putaran sebelumnya merupakan terjadi perubahan yang linier dikarenakan perbedaan pada putaran tersebut perubahannya tidak terlalu besar dan tidak terlalu kecil sehingga dapat disebut bahwa sensor level ini dapat mengukur secara linier dalam pengoperasiannya.

#### **4.1.3 Pengujian Rangkaian Jembatan *Wheatstone* dan *Analog Signal Conditioning* (ASC)**

Rangkaian penguat ini digunakan untuk mengatur tegangan masukan mikrokontroler dari sensor potensiometer. Pengujian rangkaian ini bertujuan untuk mengetahui tegangan berapa yang dihasilkan oleh rangkaian penguat yang akan digunakan masukan dari mikrokontroler *port* ADC yang kemudian dari tegangan (*analog*) tersebut dikonversi menjadi data digital, sehingga pembacaan pengukuran dari sensor potensiometer dapat memudahkan pengguna untuk mengawasi keadaan volume lumpur pada kolam lumpur (*mudtank*).

Pada pengujian rangkaian penguat ini data yang diambil sebagai variabel dari pengujian yaitu tegangan *input* (masukan) rangkaian jembatan *wheatstone*, tegangan *output* (keluaran) rangkaian jembatan *wheatstone*, tegangan *input* (masukan) rangkaian pengondisian sinyal *analog*, tegangan *output* (keluaran) rangkaian pengondisian sinyal *analog*.

Pada sub bab 3.3.2 telah dijelaskan bahwa perancangan rangkaian penguat telah ditentukan data tegangan yang telah diinginkan pada sistem ini. Maka pada Tabel 4.5 merupakan pengujian dari perancangan yang telah dilakukan untuk mengetahui hasil yang didapatkan adalah sesuai dengan hasil yang di inginkan.

**Tabel 4.5** Pengujian Rangkaian Jembatan *Wheatstone*

No.	Data tegangan yang <i>output</i> diinginkan (volt)		Tegangan <i>Input</i> (volt)	Tegangan <i>Output</i> (volt)	Keterangan (Toleransi $\pm 4\%$ )
	Min	Maks			
1.	0	2,5	4,93	2,48	V <i>Output</i> = V <i>Output</i> diinginkan
2.	0	2,5	4,93	2,51	V <i>Output</i> = V <i>Output</i> diinginkan
3.	0	2,5	4,93	2,5	V <i>Output</i> = V <i>Output</i> diinginkan

Berdasarkan pada Tabel 4.5 dapat dilihat hasil pengumpulan data dari rangkaian jembatan *wheatstone*. Dalam pengujian ini telah ditetapkan bahwa tegangan *output* yang diinginkan pada rangkaian jembatan *wheatstone* yaitu tegangan antara 0-2,5 volt. Pada pengukuran pertama rangkaian tersebut didapatkan nilai tegangan *output* sebesar 2,48 dengan tegangan *input* sebesar 5 volt, sehingga pengujian rangkaian ini telah dapat ditentukan bahwa tegangan *output* pada rangkaian jembatan *wheatstone* sama dengan tegangan *output* yang diinginkan. Keluaran tegangan ini nantinya digunakan sebagai tegangan masukan dari rangkaian pengondisian sinyal *analog*.

**Tabel 4.6** Pengujian Rangkaian ASC

No.	Data tegangan <i>output</i> yang diinginkan (volt)		Tegangan <i>Input</i> (volt)	Tegangan <i>Output</i> (volt)	Keterangan
	Min	Maks			
1.	0	5	2,48	4,96	V <i>Output</i> = V <i>Output</i> diinginkan

**Tabel 4.6** Pengujian Rangkaian ASC (lanjutan)

No.	Data tegangan <i>output</i> yang diinginkan (volt)		Tegangan <i>Input</i> (volt)	Tegangan <i>Output</i> (volt)	Keterangan
	Min	Maks			
2.	0	5	2,51	5,02	$V_{Output} = V_{Output}$ diinginkan
3.	0	5	2,5	5,00	$V_{Output} = V_{Output}$ diinginkan

Berdasarkan pada Tabel 4.6 dapat dilihat hasil pengumpulan data dari rangkaian pengondisian sinyal *analog*. Dalam pengujian ini telah ditetapkan bahwa tegangan *output* yang diinginkan pada rangkaian jembatan *wheatstone* yaitu tegangan antara 0-5 volt. Rangkaian ini mendapatkan tegangan masukan dari tegangan keluaran rangkaian jembatan *wheatstone* dan pada rangkaian ini bertujuan untuk memperkuat tegangan masukan (2,5 volt) yang kemudian dikuatkan sebanyak 2 kali untuk dapat mencapai tegangan keluaran yang diinginkan sebelumnya. Pada pengukuran pertama rangkaian tersebut didapatkan nilai tegangan *output* sebesar 4,98 volt dengan tegangan *input* sebesar 2,49 volt. Pada pengukuran kedua rangkaian tersebut didapatkan nilai tegangan *output* sebesar 5,01 volt dengan tegangan *input* sebesar 2,501 volt. Sehingga dalam pengujian rangkaian ini telah dapat ditentukan bahwa tegangan *output* pada rangkaian pengondisian sinyal *analog* sama dengan tegangan *output* yang diinginkan. Keluaran tegangan ini nantinya digunakan sebagai *input* yang ditujukan ke pin ADC dari mikrokontroler ATmega16 yang bertujuan untuk dapat mendeteksi data tegangan dikonversikan menjadi data digital (bit), sehingga dapat memudahkan pengguna untuk membaca data yang diperoleh dari sensor potensiometer.

#### 4.1.4 Pengujian Sistem Minimum ATmega16

Pengujian sistem minimum dilakukan untuk mengetahui bahwa mikrokontroler tersebut dapat berfungsi dengan baik. Pengujian ini dilakukan dengan cara memberikan suplai catu daya pada rangkaian sistem minimum ATmega16 kemudian mengecek setiap *port-port* I/O mikrokontroler, sistem *reset*, dan lampu indikator. Suatu rangkaian sistem minimum mikrokontroler dikatakan baik yaitu dengan memasukkan program pada rangkaian sistem minimum dari laptop atau komputer. Saat memasukkan program pada sistem minimum dibutuhkan perangkat perantara yang menghubungkan antara laptop/komputer dengan sistem minimum yaitu perangkat *downloader*. Selain dibutuhkan perangkat tersebut juga dibutuhkan suatu perangkat lunak *software* untuk membuat program dan mengunduh program ke sistem minimum, yaitu menggunakan *software CodeVision AVR* dan *Khazama AVR Programmer* (mengunduh program ke mikrokontroler).

Pengujian ini dilakukan dengan memasukkan sebuah program dimana seluruh *port* pada ATmega16 diberikan logika 1 (*high*) melalui program yang kemudian diunduh ke mikrokontroler. Setelah mengunduh program ke mikrokontroler memastikan tegangan *supply* untuk rangkaian sistem minimum yaitu sebesar 5 volt. Kemudian mengukur tegangan keluaran dari tiap-tiap *port* I/O pada mikrokontroler menggunakan multimeter digital. Kabel *probe* merah multimeter dihubungkan dengan kaki-kaki pada tiap *port* I/O mikrokontroler, sedangkan kabel *probe* hitam dihubungkan dengan ground. Pengukuran ini bertujuan untuk mengetahui besar tegangan dari rangkaian minimum sistem sesuai dengan batas yang dapat diterima oleh mikrokontroler. Hasil pngujian rangkaian sstem minimum mikrokontroler dapat dilihat pada Tabel 4.7.

**Tabel 4.7** Pengujian Sistem Minimum

Vin (VDC)	Port A	Vout (volt)	Port B	Vout (volt)	Port C	Vout (volt)	Port D	Vout (volt)
4,93	0	4,93	0	4,93	0	4,93	0	4,93
	1	4,93	1	4,93	1	4,93	1	4,93

**Tabel 4.7** Pengujian Sistem Minimum (lanjutan)

Vin (VDC)	Port A	Vout (volt)	Port B	Vout (volt)	Port C	Vout (volt)	Port D	Vout (volt)
	2	4,93	2	4,93	2	4,93	2	4,93
	3	4,93	3	4,93	3	4,93	3	4,93
	4	4,93	4	4,93	4	4,93	4	4,93
	5	4,93	5	4,93	5	4,93	5	4,93
	6	4,93	6	4,93	6	4,93	6	4,93
	7	4,93	7	4,93	7	4,93	7	4,93
<b>Setelah Tombol <i>RESET</i> Ditekan</b>								
4,93	0	4,93	0	4,93	0	4,93	0	4,93
	1	4,93	1	4,93	1	4,93	1	4,93
	2	4,93	2	4,93	2	4,93	2	4,93
	3	4,93	3	4,93	3	4,93	3	4,93
	4	4,93	4	4,93	4	4,93	4	4,93
	5	4,93	5	4,93	5	4,93	5	4,93
	6	4,93	6	4,93	6	4,93	6	4,93
	7	4,93	7	4,93	7	4,93	7	4,93

Besar nilai tegangan keluaran yang ditunjukkan pada Tabel 4.7 yaitu rata-rata tegangan keluaran yaitu sebesar 4,98 volt dengan batas tegangan yang dapat diterima oleh mikrokontroler yaitu dengan rentang tegangan antara 4,5 – 5,5 volt DC. Ketika keluaran tegangan lebih atau kurang dari batas yang dapat diterima oleh mikrokontroler maka mikrokontroler tidak dapat digunakan dengan baik atau tidak dapat berfungsi bahkan terjadi kerusakan pada seluruh komponen yang terhubung dengan mikrokontroler tersebut.

Dari hasil pengambilan data yang telah dilakukan, terlihat bahwa nilai tegangan yang keluar berada dibatas yang dapat diterima oleh mikrokontroler, maka mikrokontroler ATmega16 yang akan digunakan ini dalam keadaan baik atau tidak rusak.

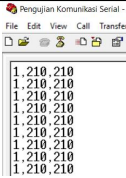
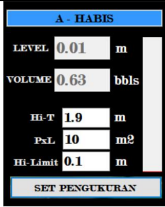
## 4.2 Pengujian Perangkat Lunak

Pengujian perangkat lunak ini meliputi pengujian pada komunikasi serial RS485 (*hyperterminal*), serta pengujian perangkat lunak yang digunakan sebagai HMI (*Human Machine Interface*) dalam segi cara penggunaan dari aplikasi tersebut.

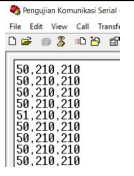
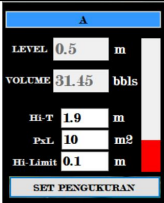
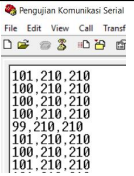
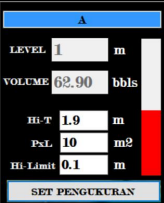
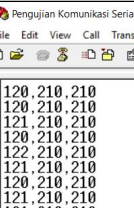
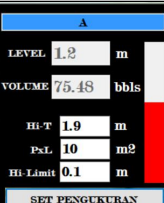
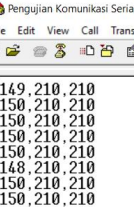
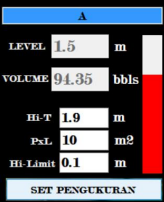
### 4.2.1 Pengujian Komunikasi Serial RS485

Rangkaian ini digunakan untuk komunikasi serial antara laptop/komputer dengan mikrokontroler. Pengujian ini dilakukan dengan mengirimkan data dari mikrokontroler (master) ke laptop/komputer (*slave*) melalui komunikasi serial. Pengujian dilakukan dengan membandingkan keluaran dari mikrokontroler menggunakan hyperterminal dengan keluaran data serial menggunakan tampilan HMI. Komunikasi serial yang digunakan hanya menggunakan komunikasi satu arah (*halfwave*) dikarenakan hanya untuk mengawasi keadaan volume dari lumpur pada kolam lumpur (*mudtank*). Hasil pengujian dapat dilihat pada Tabel 4.8.

**Tabel 4.8** Pengujian Komunikasi Serial

No.	Data yang dikirim dari mikrokontroler (cm)	Data yang diterima oleh PC	
		<i>Hyper terminal</i>	<i>HMI (Human Machine Interface)</i>
1.	1		

**Tabel 4.9** Pengujian Komunikasi Serial

No.	Data yang dikirim dari mikrokontroler (cm)	Data yang diterima oleh PC	
		<i>Hyper terminal</i>	<i>HMI (Human Machine Interface)</i>
2.	50		
3.	100		
4.	120		
5.	150		

Dari hasil data yang telah diambil dari pengujian komunikasi serial RS485 dapat disimpulkan bahwa komunikasi serial antara mikrokontroler dengan laptop (*Hyperterminal dan HMI*) telah dapat

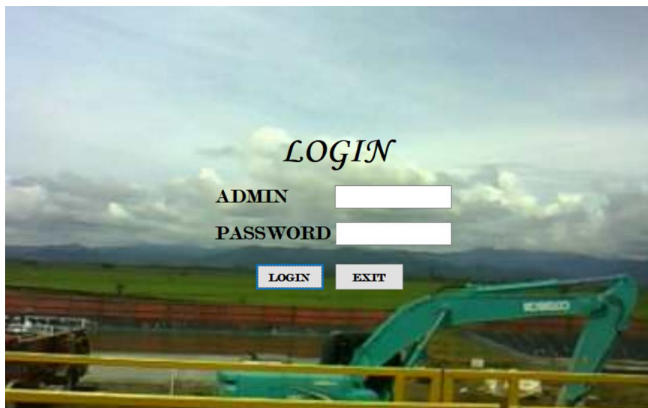
bekerja dengan baik, dibuktikan dengan hasil data yang sama dengan data yang dikirim oleh mikrokontroler dan data yang diterima oleh laptop/komputer.

#### 4.2.2 Pengujian *Human Machine Interface* (HMI)

Pada dunia industri sekarang, *Human Machine Interface* merupakan hal yang sangat penting dalam mengetahui dan mengawasi keberadaan perkembangan peralatan dari industri. HMI ini bertujuan untuk mengetahui keberhasilan dari aplikasi yang telah dibuat untuk mempermudah pekerjaan manusia dalam hal pengawasan dan *monitoring* dalam area kerja. Pada pengujian aplikasi ini digunakan untuk menampilkan hasil pembacaan dari sensor ketinggian lumpur yang kemudian dikonversikan menjadi volume.

##### 4.2.2.1 Pengujian Tampilan *Login*

Sebelum mengakses tampilan utama dalam *monitoring*, pengguna diharuskan untuk mempunyai akun yang digunakan untuk



**Gambar 4.2** Tampilan Login Monitoring

*login*. Jika pengguna tidak mempunyai akun yang benar maka orang tersebut tidak dapat mengakses atau menjalankan aplikasi tersebut. HMI hanya digunakan oleh operator tertentu untuk dapat menjalankan



*interface tersebut.* Berikut merupakan Gambar dari form login pada aplikasi tersebut.

Jika dalam pengisian admin dan *password* tersebut salah maka akan muncul peringatan “Salah dan Coba Lagi” seperti pada Gambar 4.3.



**Gambar 4.3** Tampilan *Login* Salah

Dan ketika dalam pengisian nama admin dan *password* tersebut benar maka akan muncul pesan “*Login* berhasil” dan akan



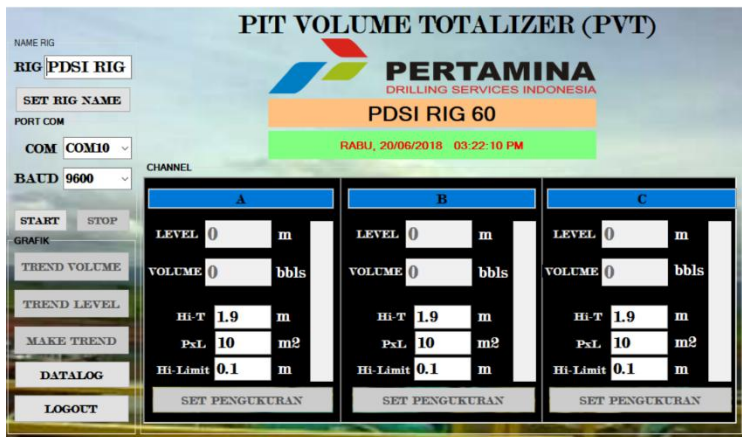
**Gambar 4.4** Tampilan *Login* Benar

menuju ke tampilan *monitoring channel* dalam mengawasi dari ketinggian lumpur. Seperti pada Gambar 4.4.

Pada pengujian tampilan login ini telah bekerja sesuai dengan algoritma perancangan pada subbab 3.4.3.2.

#### 4.2.2.2 Pengujian Tampilan *Monitoring Channel*

Pada tampilan *monitoring channel* terdapat beberapa tombol yang terbagi menjadi 4 bagian, yaitu bagian *setting* nama rig, *setting* koneksi serial, tampilan *monitoring* ketinggian, *monitoring* grafik dan penyimpanan *database*. *Setting* nama rig digunakan untuk mengatur nama rig yang akan digunakan pada saat pengeboran. *Setting* koneksi serial digunakan untuk mengatur sambungan koneksi serial yang terdapat dalam komputer, Tampilan *monitoring* ketinggian digunakan untuk melihat dan mengawasi ketinggian dan volume dari kolam lumpur (*mudtank*). *Monitoring* grafik digunakan untuk melihat grafik yang keluar ketika ketinggian dan volume tersebut terukur. Pada



Gambar 4.5 Tampilan *Monitoring Channel*

*monitoring channel* ini terdapat tampilan data ketinggian dan volume keadaan lumpur pada *mudtank*. Selanjutnya data tersebut masuk ke dalam penyimpanan *database* sebagai pencatat *history* dari pemakaian

sistem *monitoring* ini. *Database* yang kami gunakan yaitu *database* MySQL. Berikut pada Gambar 4.5 merupakan tampilan *monitoring channel*.

#### 4.2.2.3 Pengujian Pengatur Nama Rig

Pengatur nama rig adalah suatu kegiatan yang memberikan nama pada rig yang digunakan selama proses pengeboran. Pengujian pengaturan nama rig ini bertujuan untuk mengetahui bahwa saat pengisian nama rig dapat ditampilkan pada persegi panjang yang berwarna orange. Nama rig tersebut merupakan bentuk identitas alat



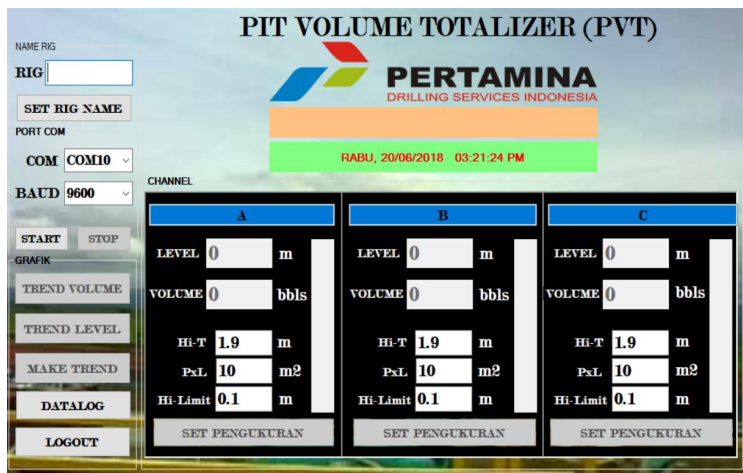
NAME RIG

RIG

SET RIG NAME

Gambar 4.6 Tombol Pengatur Nama Rig

dari pengeboran. Langkah-langkah mengatur nama rig tersebut yaitu pertama menuliskan nama rig yang digunakan pada kotak teks RIG



PIT VOLUME TOTALIZER (PVT)

PERTAMINA  
DRILLING SERVICES INDONESIA

RABU, 20/06/2018 03:21:24 PM

NAME RIG

RIG

SET RIG NAME

PORT COM

COM COM10

BAUD 9600

CHANNEL

	A	B	C
LEVEL	0 m	0 m	0 m
VOLUME	0 bbls	0 bbls	0 bbls
Hi-T	1.9 m	1.9 m	1.9 m
Ps.L	10 m <sup>2</sup>	10 m <sup>2</sup>	10 m <sup>2</sup>
Hi-Limit	0.1 m	0.1 m	0.1 m
SET PENGUKURAN	SET PENGUKURAN	SET PENGUKURAN	SET PENGUKURAN

START STOP

GRAFIK

TREND VOLUME

TREND LEVEL

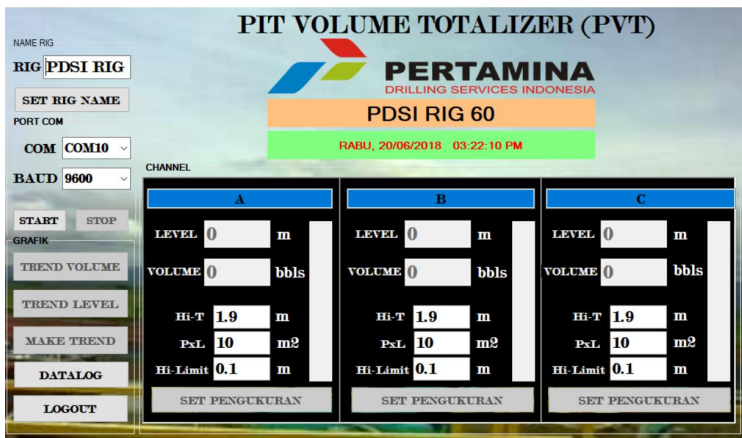
MAKE TREND

DATALOG

LOGOUT

Gambar 4.7 Tampilan Sebelum Mengatur Nama Rig

(Gambar 4.6), kotak tersebut dapat menuliskan nama apapun yang ingin dimasukkan sebagai identitas tersebut. Setelah menuliskan nama rig maka klik tombol “*SET RIG NAME*” sebanyak 1 kali maka akan muncul teks yang telah ditulis tadi pada kotak berwarna orange (Gambar 4.8), sedangkan Gambar 4.7 sebelum mengalami perubahan nama rig. Seperti pada Gambar 4.6 merupakan tombol yang dapat menampilkan nama rig yang telah dituliskan



**Gambar 4.8** Tampilan Sesudah Mengatur Nama Rig

#### 4.2.2.4 Pengujian Koneksi Serial

Pada pengujian koneksi serial ini digunakan untuk menentukan *port* sambungan dari komunikasi serial RS485 to USB yang telah terintegrasi dengan mikrokontroler sebagai perantara dari sensor ke laptop/komputer/HMI. Apabila penentuan *port* tersebut tidak sesuai dengan sambungan dari mikrokontroler tersebut maka pembacaan nilai dari sensor tersebut tidak akan terbaca pada HMI. Pada pengujian kali ini kami menggunakan *port* com sambungan yaitu COM10 (COM mikrokontroler). Langkah-langkah penyambungan *port* yaitu setelah melakukan pengaturan nama rig maka mengatur sambungan *port* serial dengan memeriksa sambungan pada “*device manager*” laptop. Ketika terdapat sambungan *port* serial maka pada HMI akan otomatis memilih

untuk sambungan serial yang teratas sendiri dari laptop (*port com* harus sesuai dengan *port* serial). Kemudian memilih *baudrate* yang digunakan pada HMI, pemilihan *baudrate* pada HMI harus sama dengan *baudrate* dari mikrokontroler (Gambar 4.9). Setelah diperiksa sambungan *port* dan *baudrate* sudah benar lalu klik tombol *start* untuk memulai pembacaan nilai dari sensor tersebut terlihat pada Gambar 4.10.



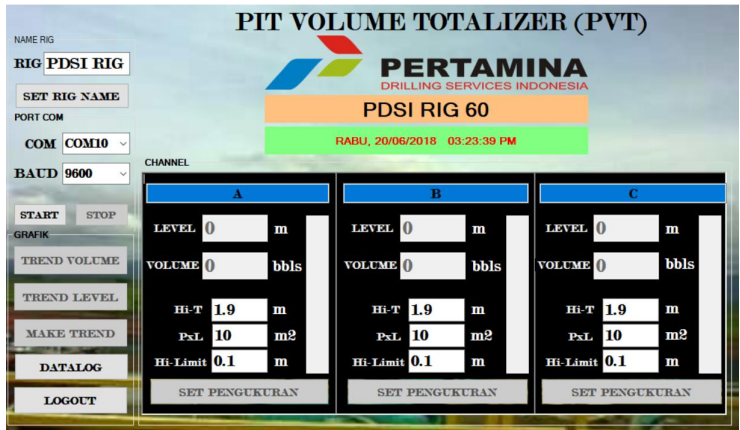
Gambar 4.9 Pengatur Sambungan *Port Com*



Gambar 4.10 Tampilan Setelah Tombol *Start* Ditekan

Pengujian selanjutnya yaitu memutuskan sambungan dari *port* serial dengan HMI yaitu dengan menekan tombol “STOP” pada bagian pengatur *Port COM* maka sambungan *port* serial akan terputus yang

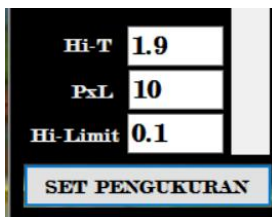
menyebabkan pembacaan dari nilai sensor tersebut telah dihentikan. Berikut tampilan saat tombol *STOP* ditekan yang berfungsi untuk memutuskan sambungan *port* serial.



Gambar 4.11 Tampilan Saat Pemutusan *Port* Serial

#### 4.2.2.5 Pengujian Penentuan Luas Media yang dipakai

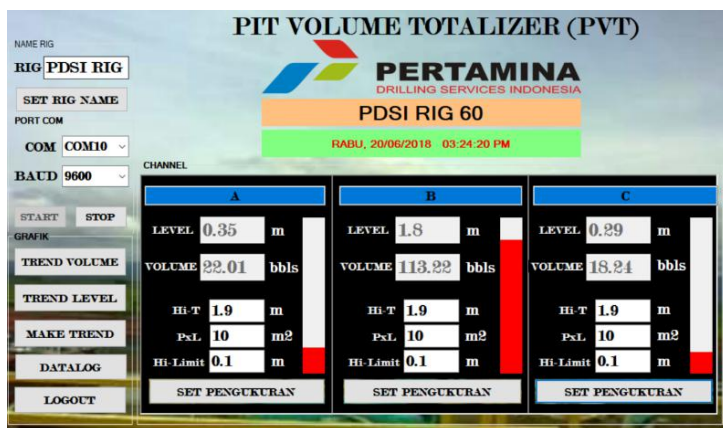
Pada pengujian ini bertujuan untuk memasukkan luas media yang dipakai saat pengukuran ketinggian lumpur. Cara menguji penentuan luas yaitu dengan cara mengkalikan luas yang telah dimasukkan dengan ketinggian yang telah terdeteksi oleh sensor, sehingga menghasilkan volume yang sesuai dengan ukuran dari media yang dipakai, akan tetapi dalam pengukuran volume tersebut telah dikonversi menjadi satuan barel (bbls).



Gambar 4.12 Pengaturan Luas, Hi-Top, dan Hi-Limit

#### 4.2.2.6 Pengujian Data Ketinggian dan Volume dari Mikrokontroler

Dalam pengujian data ketinggian dari mikrokontroler ini bertujuan untuk mengetahui kerja dari tampilan *channel* ini berfungsi dengan baik atau tidak. Langkah-langkah untuk menguji tampilan *channel*, pertama menentukan batas ketinggian saat keadaan penuh dan



Gambar 4.14 Pengukuran Data Ketinggian ke-1

saat keadaan kosong serta menentukan luas media yang dipakai saat pengukuran (Gambar 4.13). Kemudian klik tombol set pengukuran untuk memulai pembacaan data nilai ketinggian dan volume (Gambar



Gambar 4.13 Pengukuran Data Ketinggian 2

4.14). Jika terdapat salah satu atau lebih *channel* mencapai batas keadaan penuh ataupun kosong maka pada kotak teks *channel* berwarna biru saat keadaan penuh akan menampilkan teks “PENUH”, dan jika dalam keadaan kosong maka akan menampilkan teks “HABIS”. (Gambar 4.14). Sedangkan pada animasi yang berwarna merah tersebut menandakan keadaan tangki, ketika keadaan penuh maka animasi tersebut akan berwarna merah penuh, ketika keadaan kosong maka animasi tersebut akan berwarna putih. Animasi tersebut menandakan untuk keadaan tangki yang ada dilapangan/nyata (Gambar 4.14).

#### 4.2.2.7 Pengujian Tampilan Grafik

Dalam pengujian tampilan grafik bertujuan untuk mengetahui keadaan level dan volume secara realtime untuk mengetahui perubahan dari level dan volume secara berkala. Langkah-langkah menguji tampilan grafik tersebut yaitu pertama memastikan bahwa sambungan *port* serial telah tersambung dan telah melakukan pembacaan data nilai level dan volume. Kedua, memastikan bahwa *channel* yang akan diukur telah ditekan tombol “SET PENGUKURAN”. Ketiga, yaitu menekan tombol “TREND VOLUME” atau “TREND LEVEL” atau “MAKE TREND” (Gambar 4.15). Pada tombol *trend volume* ketika



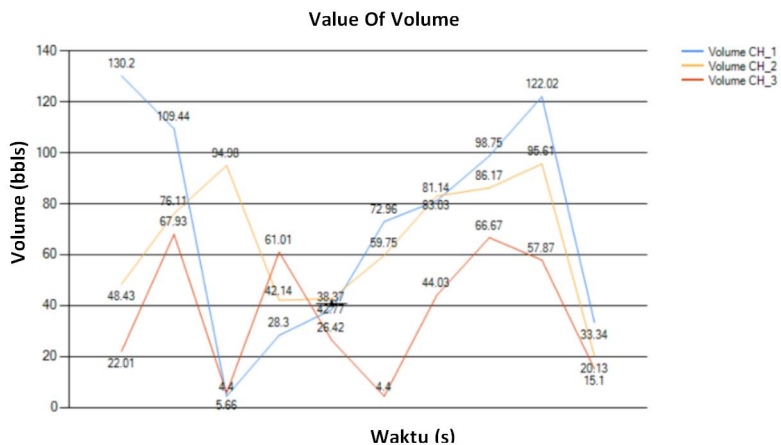
Gambar 4.15 Tombol Menu Grafik

ditekan, maka akan memunculkan grafik hanya pengukuran volume

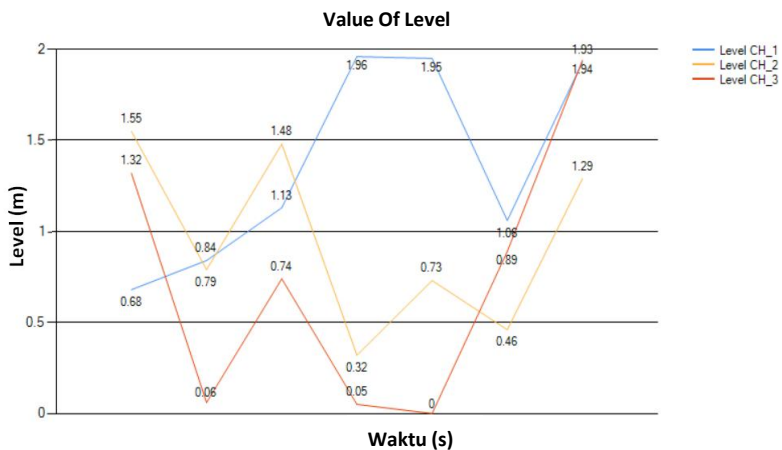


(Gambar 4.16), ketika *trend level* ditekan, maka akan memunculkan grafik hanya pengukuran *level* (Gambar 4.17), sedangkan ketika *make trend* ditekan, maka akan menampilkan grafik rata-rata nilai dari *level* dan *volume* yang telah terukur pada sistem tersebut (Lampiran B-1).

Grafik tersebut bertujuan untuk mengetahui dan menunjukkan kondisi *volume lumpur* yang sebelumnya telah diukur.



Gambar 4.16 Tampilan Grafik *Volume*



**Gambar 4.17** Tampilan Grafik *Level*

#### 4.2.2.8 Pengujian Penyimpanan *Database*

*Database* merupakan komponen penting dalam hal *monitoring* atau pengawasan dikarenakan *monitoring* membutuhkan data-data yang telah dilakukan sebelumnya untuk mengetahui riwayat penggunaan dari sistem tersebut agar dapat dilakukan perawatan. Langkah-langkah pengujian penyimpanan *database* yaitu pertama memastikan bahwa sambungan *port* serial telah tersambung dan telah melakukan pembacaan data nilai *level* dan volume. Kedua, memastikan bahwa *channel* yang akan diukur telah ditekan tombol “SET PENGUKURAN”. Ketika tombol tersebut ditekan maka penyimpanan data yang terbaca akan tersimpan pada *database*. Jika ingin mengetahui riwayat dari pengukuran sistem tersebut yaitu dengan menekan tombol “DATALOG” yang berfungsi untuk membuka penyimpanan *database* tersebut. Saat menekan tombol “DATALOG” maka akan menampilkan form pengambilan data yang telah dilakukan (Gambar 4.18).



**Gambar 4.18** Tampilan Tombol Menu Datalog

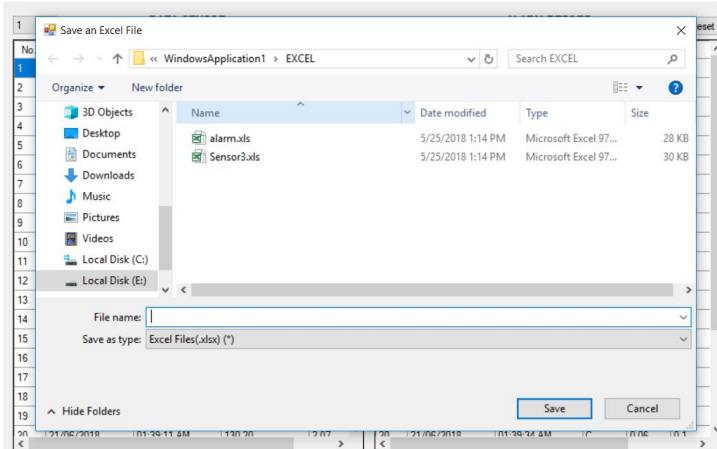
Dalam penyimpanan tersebut terbagi menjadi 2 bagian, yaitu penyimpanan data yang terukur per 1 detik dan penyimpanan data yang terukur saat dalam keadaan penuh atau kosong. Dalam penyimpanan data per 1 detik data yang tersimpan yaitu berupa data volume, data level, tanggal dan waktu pengukuran. Sedangkan penyimpanan data saat keadaan penuh atau kosong yaitu berupa batas ketinggian volume

yang telah ditentukan, hasil yang terukur, dan *channel* saat keadaan penuh atau kosong (Gambar 4.19).

DATA SENSOR					ALARM RECORD					
No.	Date	Time	Volume	Level	No.	Date	Time	PIT	Value	SP
1	25/05/2018	01:06:15 PM	4.40	0.07	1	25/05/2018	01:14:39 PM	A	1.9	1.9
2	25/05/2018	01:06:16 PM	71.70	1.14	2	25/05/2018	01:14:42 PM	A	1.93	1.9
3	25/05/2018	01:06:17 PM	26.42	0.42	3	25/05/2018	01:14:45 PM	A	2.06	1.9
4	25/05/2018	01:06:18 PM	81.77	1.3	4	25/05/2018	01:14:47 PM	A	0.04	0.1
5	25/05/2018	01:06:19 PM	119.51	1.9	5	25/05/2018	01:14:49 PM	A	1.9	1.9
6	25/05/2018	01:06:20 PM	42.14	0.67	6	25/05/2018	01:14:49 PM	A	0.09	0.1
7	25/05/2018	01:06:21 PM	30.82	0.49	7	25/05/2018	01:14:50 PM	A	0.1	0.1
8	25/05/2018	01:06:22 PM	83.65	1.33	8	25/05/2018	01:15:05 PM	A	1.94	1.9
9	25/05/2018	01:06:23 PM	83.65	1.33	9	25/05/2018	01:15:05 PM	A	2.02	1.9
10	25/05/2018	01:06:24 PM	108.81	1.73	10	25/05/2018	01:15:14 PM	A	0.03	0.1
11	25/05/2018	01:06:25 PM	35.22	0.56	11	25/05/2018	01:15:18 PM	A	2.07	1.9
12	25/05/2018	01:06:26 PM	103.15	1.64	12	25/05/2018	01:15:19 PM	A	0.05	0.1
13	25/05/2018	01:06:27 PM	45.92	0.73	13	25/05/2018	01:15:21 PM	A	2.03	1.9
14	25/05/2018	01:06:28 PM	40.25	0.64	14	25/05/2018	01:15:22 PM	A	0.02	0.1
15	25/05/2018	01:06:29 PM	59.12	0.94	15	25/05/2018	01:15:26 PM	A	0.04	0.1
16	25/05/2018	01:06:30 PM	103.15	1.64	16	25/05/2018	01:15:33 PM	A	0.06	0.1
17	25/05/2018	01:06:31 PM	89.32	1.42	17	25/05/2018	01:15:36 PM	A	0.04	0.1
18	25/05/2018	01:06:32 PM	15.10	0.24	18	25/05/2018	01:15:58 PM	C	1.96	1.9
19	25/05/2018	01:06:33 PM	61.64	0.98	19	25/05/2018	01:15:59 PM	B	1.99	1.9
20	25/05/2018	01:06:34 PM	120.14	1.91	20	25/05/2018	01:15:59 PM	C	0	0.1

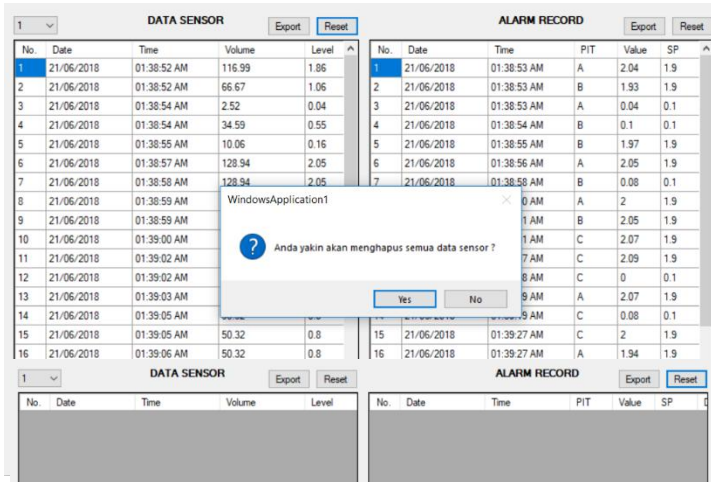
Gambar 4.19 Tampilan Penyimpanan Database

Ketika ingin mengambil data yang telah terukur tersebut, maka tekan tombol *export* pada tampilan penyimpanan tersebut untuk menyimpan file dalam bentuk excel (.xls).

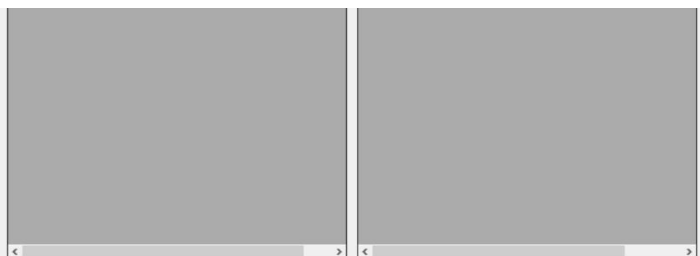


**Gambar 4.20** Penyimpanan ke Excel (.xls)

Ketika ingin menghapus penyimpanan yang terdapat dalam database tersebut, maka tekan tombol *reset* pada tampilan tersebut.



**Gambar 4.21** Konfirmasi Penghapusan Data Sensor



**Gambar 4.22** Tampilan Saat Data Sensor Telah Dihapus

Kemudian pilih “YES” untuk mengkonfirmasi penghapusan data sensor. Setelah mengkonfirmasi penghapusan tersebut, maka tampilan

DATA SENSOR					ALARM RECORD					
No.	Date	Time	Volume	Level	No.	Date	Time	PIT	Value	SP
1	21/06/2018	01:46:40 AM	45.92	0.73						
2	21/06/2018	01:46:41 AM	96.23	1.53						
3	21/06/2018	01:46:42 AM	81.77	1.3						
4	21/06/2018	01:46:43 AM	40.25	0.64						

Gambar 4.23 Tampilan Sebelum Ditekan Refresh

penyimpanan akan seperti Gambar 4.21.

Ketika ingin mengetahui pengukuran yang dilakukan setelah melakukan penyimpanan, maka tekan tombol *refresh* untuk mengetahui pengukuran yang dilakukan. Pada Gambar 4.24 merupakan tampilan sebelum ditekan tombol *refresh*. Sedangkan pada Gambar 4.25 merupakan tampilan sesudah menekan tombol *refresh*.

### 4.3 Pengujian Keseluruhan Sistem.

DATA SENSOR					ALARM RECORD					
No.	Date	Time	Volume	Level	No.	Date	Time	PIT	Value	SP
1	21/06/2018	01:46:40 AM	45.92	0.73	1	21/06/2018	01:46:46 AM	C	0.1	0.1
2	21/06/2018	01:46:41 AM	96.23	1.53	2	21/06/2018	01:46:47 AM	A	1.9	1.9
3	21/06/2018	01:46:42 AM	81.77	1.3	3	21/06/2018	01:46:50 AM	B	2.02	1.9
4	21/06/2018	01:46:43 AM	40.25	0.64	4	21/06/2018	01:46:50 AM	A	1.93	1.9
5	21/06/2018	01:46:44 AM	94.98	1.51	5	21/06/2018	01:46:51 AM	A	0.01	0.1
6	21/06/2018	01:46:45 AM	104.41	1.66	6	21/06/2018	01:46:51 AM	B	1.94	1.9
7	21/06/2018	01:46:46 AM	51.58	0.82	7	21/06/2018	01:46:51 AM	C	1.92	1.9
8	21/06/2018	01:46:47 AM	27.05	0.43	8	21/06/2018	01:46:52 AM	A	0.08	0.1
9	21/06/2018	01:46:48 AM	119.51	1.9	9	21/06/2018	01:46:53 AM	A	2.06	1.9
10	21/06/2018	01:46:49 AM	34.59	0.55	10	21/06/2018	01:46:54 AM	C	0.08	0.1
11	21/06/2018	01:46:50 AM	34.59	0.55	11	21/06/2018	01:46:55 AM	A	0.04	0.1
12	21/06/2018	01:46:51 AM	121.39	1.93	12	21/06/2018	01:46:56 AM	B	0.05	0.1
13	21/06/2018	01:46:52 AM	16.98	0.27	13	21/06/2018	01:46:56 AM	B	2	1.9
14	21/06/2018	01:46:53 AM	5.03	0.08	14	21/06/2018	01:46:57 AM	A	1.9	1.9
15	21/06/2018	01:46:54 AM	129.57	2.06	15	21/06/2018	01:46:57 AM	A	0.09	0.1
16	21/06/2018	01:46:55 AM	27.05	0.43	16	21/06/2018	01:46:58 AM	A	0.1	0.1
17	21/06/2018	01:46:56 AM	2.52	0.04	17	21/06/2018	01:46:59 AM	C	1.9	1.9
18	21/06/2018	01:46:57 AM	11.95	0.19	18	21/06/2018	01:46:59 AM	B	1.94	1.9
19	21/06/2018	01:46:58 AM	5.66	0.09	19	21/06/2018	01:47:00 AM	A	0.08	0.1
20	21/06/2018	01:46:59 AM	6.28	0.1	20	21/06/2018	01:47:01 AM	C	1.91	1.9

Gambar 4.24 Tampilan Sesudah Ditekan Refresh

Pengujian keseluruhan dilakukan dengan melakukan pengukuran jarak dengan menggunakan pelampung dengan berbagai metode yaitu pengukuran manual tanpa pelampung, dan pengukuran pada ketinggian permukaan air dalam kolam bak untuk membuktikan bahwa perubahan yang terjadi mendekati dengan kenyataan. Pengujian ini dilakukan dengan menghubungkan sensor potensiometer ke sistem mikrokontroler ATmega16, dan melihat tampilan hasil pengukuran pada HMI. Sehingga sensor dapat mendeteksi ketika adanya gerakan kenaikan atau turunnya air.



**Gambar 4.25** Pengujian Sistem Keseluruhan 1

Hasil pengukuran ketinggian ditampilkan pada Tabel 4.9 untuk pengukuran manual dengan perbandingan HMI yang dilakukan sebanyak 3 kali untuk mendapatkan nilai yang presisi (diukur beberapa kali tetap menunjukkan nilai yang sama) dan akurat (mendekati nilai sebenarnya). Pengukuran secara manual dilakukan dengan cara memutar potensiometer sambil diukur setiap 1 kali putaran yaitu 21 cm dan dibandingkan dengan hasil dari data ketinggian nyata dengan hasil yang terdapat pada HMI.

**Tabel 4.10** Pengukuran Ketinggian Manual

Putaran	Ketinggian Tandon	Pengukuran ke-1 (cm)	Pengukuran ke-2 (cm)	Pengukuran ke-3 (cm)	Nilai Rata-rata (cm)

1.	21	19,3	20,1	21,5	20,625
2.	42	42,6	39,9	43,1	42,075
3.	63	58,4	62,5	60,1	60,6
4.	84	80,1	78,3	81,0	80,225
5.	105	103,1	102,3	101,5	102,325
6.	126	126,2	124,7	125,8	125,475
7.	147	138,9	142,4	121,5	131,175
8.	168	159,9	158,2	160,5	159,3
9.	189	189,1	189,8	188,3	189,175
10.	210	210	210	210	210

Berdasarkan Tabel 4.9 didapatkan data hasil pengukuran secara manual serta didapatkan nilai rata-rata dari ketinggian yang terukur. Dari pengukuran tersebut terdapat perbedaan antara pengukuran ketinggian nyata dengan ketinggian yang telah terukur.

$$Error = \frac{Ketinggian Tandon - Rata ketinggian diukur}{Ketinggian Tandon} \times 100\% \dots \dots \dots (4.1)$$

$$Error = \frac{21 - 20.625}{21} \times 100\% \dots \dots \dots (4.2)$$

$$Error = 1,9 \% \dots \dots \dots (4.3)$$

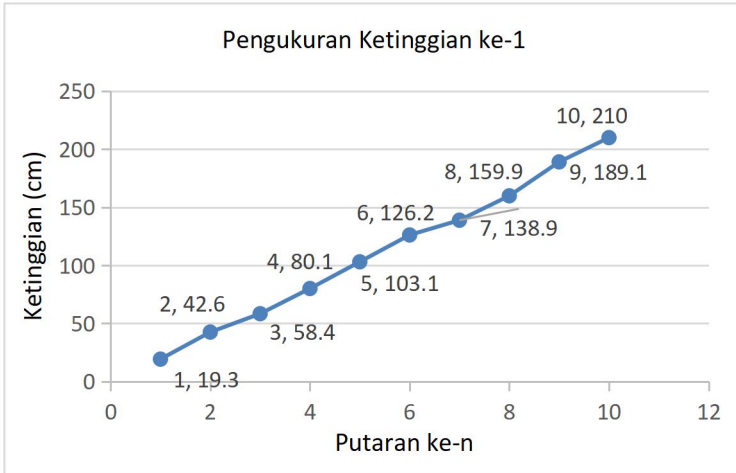
$$Error = \frac{105 - 102,35}{105} \times 100\% \dots \dots \dots (4.4)$$

$$Error = 2,52 \% \dots \dots \dots (4.5)$$

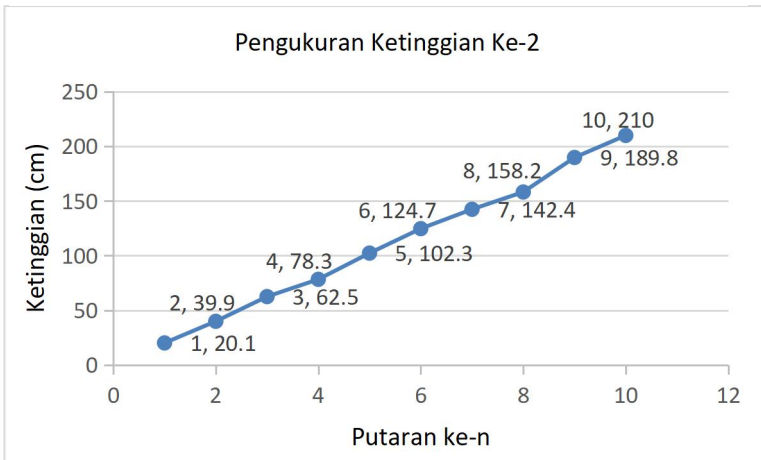
$$Error = \frac{210 - 210}{210} \times 100\% \dots \dots \dots (4.6)$$

$$Error = 0 \% \dots \dots \dots (4.7)$$

Dari perbandingan data tersebut didapatkan bahwa kesalahan dalam pengukuran ketinggian dan volume yaitu sebesar  $\pm 0,85\%$ . Dapat dihasilkan grafik linier saat dilakukan pengukuran menggunakan



**Gambar 4.27** Pengukuran Ketinggian Ke-1



**Gambar 4.26** Pengukuran Ketinggian Ke-2



sistem tersebut. Grafik linier tersebut menunjukkan bahwa data yang diperoleh dalam melakukan pengukuran 1 hingga ke-n selisih perbedaannya mendekati sama, sehingga pengukuran ketinggian dan volume lumpur pada *mudtank* memiliki tingkat keakuratan sebesar  $\pm 99,15\%$ , dengan melihat dari *error* yang didapatkan saat pengukuran.

**Tabel 4.11** Pengukuran Volume Manual

Putaran	Volume Tandon (bbls)	Pengukuran ke-1 (bbls)	Pengukuran ke-2 (bbls)	Pengukuran ke-3 (bbls)	Nilai Rata-rata (bbls)
1.	4,75	4,68	4,8	4,72	4,7325
2.	9,5	9,1	9,3	9,41	9,315
3.	14,25	14,05	14,23	14,16	14,14
4.	19	18,7	19,01	18,95	18,8875
5.	23,75	22,6	23,4	23,53	23,24
6.	28,5	28,1	28,64	28,49	28,4425
7.	33,25	32,9	33,05	33,15	32,925
8.	38	38,03	37,78	37,78	37,88
9.	42,75	41,5	40,97	42,6	0
10.	47,5	46,9	46,9	46,9	0

Dalam pengukuran ini menggunakan media tandon yang berukuran panjang 2 meter, lebar 1,8 meter, dan tinggi 2 meter.

Berdasarkan Tabel 4.10 didapatkan data hasil pengukuran secara manual serta didapatkan nilai rata-rata dari volume yang terukur. Dari pengukuran tersebut terdapat perbedaan antara pengukuran volume nyata dengan volume yang telah terukur.

$$Error = \frac{Volume\ Tandon - Rata\ volume\ diukur}{Volume\ Tandon} \times 100\% \dots \dots \dots (4.8)$$

$$Error = \frac{4,75 - 4,7325}{4,75} \times 100\% \dots \dots \dots (4.9)$$

$$Error = 0,36\% \dots \dots \dots (4.10)$$

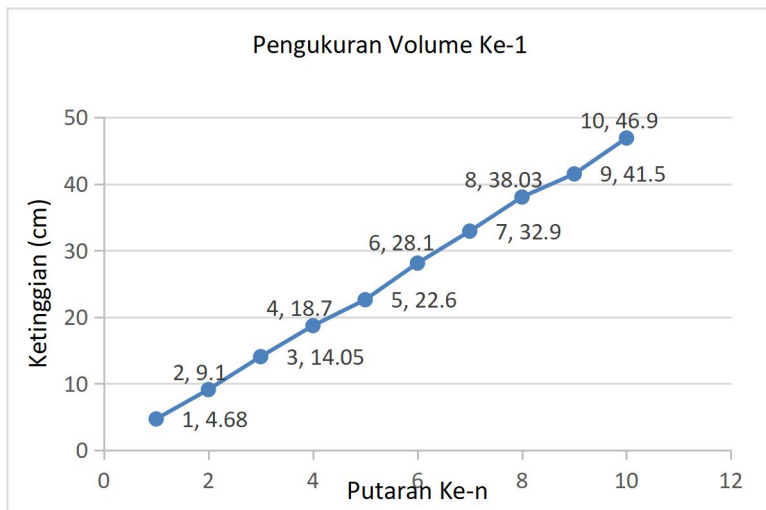
$$Error = \frac{23,75-23,24}{23,75} \times 100\% \dots \dots \dots (4.11)$$

$$Error = 2,14 \% \dots \dots \dots (4.12)$$

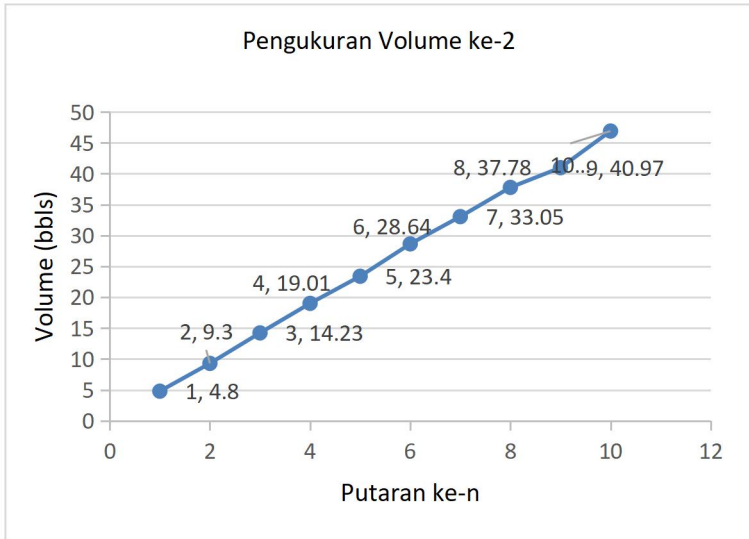
$$Error = \frac{38-37,88}{38} \times 100\% \dots \dots \dots (4.14)$$

$$Error = 0,31 \% \dots \dots \dots (4.15)$$

Dari perbandingan data tersebut didapatkan bahwa kesalahan dalam pengukuran volume yaitu sebesar ± 0,85%, dapat dihasilkan grafik linier saat dilakukan pengukuran menggunakan sistem tersebut. Grafik linier tersebut menunjukkan bahwa data yang diperoleh dalam melakukan pengukuran 1 hingga ke-n selisih perbedaannya mendekati sama, sehingga pengukuran ketinggian dan volume lumpur pada *mudtank* memiliki tingkat keakuratan sebesar ± 99,15 %, dengan melihat dari *error* yang didapatkan saat pengukuran.



**Gambar 4.28** Pengukuran Volume Ke-1



**Gambar 4.29** Pengukuran Volume Ke-2

## **BAB V**

### **PENUTUP**

Dari proses perancangan alat dan pengambilan data yang kami lakukan, terdapat beberapa kesimpulan yang dapat kami ambil. Selain itu terdapat juga beberapa saran untuk penelitian selanjutnya agar dapat diperbaiki dan dikembangkan sehingga dapat lebih baik dan lebih bermanfaat. Berdasarkan data dan hasil dari pengujian serta analisa data dari Rancang Bangun Sistem *Monitoring* Volume Lumpur pada *Mudtank* Berbasis Mikrokontroler didapatkan tingkat keakuratan pengukuran ketinggian dan volume pada sistem ini yaitu sebesar  $\pm 99,15 \%$ , sehingga dapat disimpulkan bahwa lumpur yang akan dimasukkan kedalam lubang pengeboran akan mencukupi kebutuhan sirkulasi lumpur pada saat pengeboran berlangsung, tidak mengalami kelebihan dan kekurangan volume lumpur, serta dapat mempermudah operator untuk mengetahui dan mengawasi keadaan volume yang terdapat dalam *mudtank*. Dalam pengerjaan terdapat kendala untuk mengerjakan tugas akhir ini yaitu pembuatan mekanik sensor ketinggian, sehingga untuk pengembangan dan penyempurnaan pembuatan mekanik tersebut dapat diperbaiki dan dirapikan dengan menggunakan sistem katrol yang digerakkan oleh motor. Selain itu, juga dapat dikembangkan dengan menggunakan data yang tersimpan dapat dikirimkan langsung ke pusat kendali melalui media *wireless*, modem, atau bahkan dikirim melalui *website*, dan dapat memberikan peringatan ketika kolam lumpur dalam keadaan penuh atau kosong dengan menambahkan indikator *buzzer* secara otomatis.

-----Halaman ini sengaja dikosongkan-----

## DAFTAR PUSTAKA

- [1] Kaswir, Badu. 2008, “*Drilling Fluids and Hydraulics*”, Diklat Advance Drilling, Pusdiklat Migas.
- [2] Amin M. Mustaghfirin. 2013, “Lumpur dan Hidrolika Lumpur Pengeboran, Jakarta, Kementerian Pendidikan dan Kebudayaan Republik Indonesia.
- [3] Nurfitri. Leila, 2014, “Pengukur Level Ketinggian Cairan Pada Kolam Berbasis Mikrokontroler ATmega8”, Laporan Proyek Akhir, Fakultas Teknik, Jurusan Teknik Elektro, Jogjakarta, Universitas Gadjah Mada.
- [4] Budiharto, Widodo, 2010, “Robotika : Teori + Implementasi”, Yogyakarta, Andi
- [5] Lamsani, Missa, 2012, “Pengondisian Sinyal”, Depok, Universitas Gunadharma.
- [6] \_\_\_\_\_, 2018, “Rangkaian Jembatan dan Potensiometer”, <https://nanopdf.com/download/31-rangkaian-jembatan-dan-potensiometer-rangkaian.pdf> (diakses pada tanggal 23 April 2018)
- [7] Shrader, Robert L, 1991, “Komunikasi Elektronika”, Jakarta, Erlangga.
- [8] Kho, Dickson. 2015, “Prinsip Kerja DC *Power supply* (Adaptor)”.<http://teknikelektronika.com/prinsip-kerja-dc-power-supply-adaptor/> (diakses pada tanggal 24 April 2018).
- [9] Wildan, Ali. 2014, “Komunikasi RS485 *Multipoint*”, Bandung, Unikom.
- [10] Akbar, Ali, 2005, “*Visual Basic .Net* Belajar Praktis Melalui Berbagai Tutorial dan Tips, Bandung, Informatika.
- [11] Priliani, R Ariesta, 2014, “Rancang Bangun Sistem Keamanan Ruang Menggunakan *Radio Frequency Identification* (RFID) Berbasis *Wavecom* M1206B”, Teknik Elektronika, Palembang, Politeknik Sriwijaya.
- [12] Nugroho, Bunafit. 2013. Dasar Pemrograman Web PHP – MySQL dengan Dreamweaver. Yogyakarta : Gava Media
- [13] Asmawati. D, 2016, “Sistem Informasi Auntansi Penerimaan

dan Pengeluaran Kas Berbasis *Web* pada UIN Raden Fatah Palembang”, Palembang, UIN Raden Fatah.

- [14] Anggit, 2016, “Membuat *Database* di XAMPP”, <https://www.rumahweb.com/journal/membuat-database-di-xampp.htm> (diakses pada tanggal 15 Mei 2018)





```

// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}

long map(long x, long in_min, long in_max, long out_min, long
out_max)
{
    if ((x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min
< out_min){return out_min;}
    else if ((x - in_min) * (out_max - out_min) / (in_max - in_min) +
out_min > out_max){return out_max;}
    else {return (x - in_min) * (out_max - out_min) / (in_max - in_min) +
out_min;}
}

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

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTA=0xFF;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTB=0x00;

```

```
DDRB=0x00;
```

```
// Port C initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In  
Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T  
State0=T
```

```
PORTC=0x00;
```

```
DDRC=0x00;
```

```
// Port D initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In  
Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T  
State0=T
```

```
PORTD=0x00;
```

```
DDRD=0x00;
```

```
// Timer/Counter 0 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer 0 Stopped
```

```
// Mode: Normal top=0xFF
```

```
// OC0 output: Disconnected
```

```
TCCR0=0x00;
```

```
TCNT0=0x00;
```

```
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer1 Stopped
```

```
// Mode: Normal top=0xFFFF
```

```
// OC1A output: Discon.
```

```
// OC1B output: Discon.
```

```
// 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=0x00;
TCCR1B=0x00;
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=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
```

```
// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
```

```
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: Off
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x08;
```

```

UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x4D;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 750,000 kHz
// ADC Voltage Reference: AREF pin
// ADC Auto Trigger Source: ADC Stopped
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

while (1)
{
    //printf("%d",read_adc(1));
    printf("%d",map(read_adc(0), 5, 1017, 0, 210));
    printf("%d",map(read_adc(1), 16, 1021, 0, 210));
    printf("%d",map(read_adc(2), 5, 1017, 0, 210));
    printf("\n\r");
    delay_ms(500);
}
}

```

### **A-2. Listing Program Visual Basic Form Login.**

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
        If TextBox1.Text = "admin" And TextBox2.Text =
"admin" Then
            MessageBox.Show("Berhasil", "Log In",
MessageBoxButtons.OK)
            TextBox1.Text = ""
            TextBox2.Text = ""
            Me.Hide()
            Form2.Show()
        Else
            MsgBox("Salah dan Coba Lagi",
MsgBoxStyle.OkOnly, "Invalid")
        End If
    End Sub
    Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
        Me.Close()
    End Sub

    Private Sub Form1_Load(sender As Object, e As
EventArgs) Handles MyBase.Load
    End Sub
End Class
```

### **A-3. Listing Program Visual Basic Form Channel.**

```
Imports System
Imports System.Windows.Forms
Imports System.Threading
Imports System.IO.Ports
Imports System.ComponentModel
Imports
System.Windows.Forms.DataVisualization.Charting
Imports MySql.Data.MySqlClient

Public Class Form2
```

```

    Delegate Sub SetTextCallBack(ByVal [text] As
String)
    Dim engChart As New
System.Windows.Forms.DataVisualization.Charting.Series

    Private Sub btnRig_Click(sender As Object, e As
EventArgs) Handles btnRig.Click
        Label1.Text = txtRig.Text.ToUpper
    End Sub

    Private Sub btnExit_Click(sender As Object, e As
EventArgs) Handles btnExit.Click
        Dim Ingin As MsgBoxResult
        Ingin = MsgBox("Apakah Anda akan keluar?",
MsgBoxStyle.YesNo, "Keluar")
        If Ingin = MsgBoxResult.Yes Then
            If SerialPort1.IsOpen Then
                SerialPort1.Close()
            End If
            reset()
            Clock.Stop()
            Form1.Show()
            Me.Hide()
        ElseIf Ingin = MsgBoxResult.No Then
            Exit Sub
        End If
    End Sub

    Dim myPort As Array 'COM Ports detected on the
system will be stored here
    Dim y As Integer = 0
    Sub serialKonek()
        myPort = IO.Ports.SerialPort.GetPortNames()
'Get all com ports available
        For Me.y = 0 To UBound(myPort)
            cmbPort.Items.Add(myPort(y))
        Next
        If myPort.Length = 0 Then
            MessageBox.Show("Perangkat tidak
terhubung...")

```

```

        btnStart.Enabled = true
        btnStop.Enabled = False
        btnLevel.Enabled = False
        btnVolume.Enabled = False
        btnMaketrend.Enabled = False
        btnDatalog.Enabled = True
        btnExit.Enabled = True
        btnSET2.Enabled = False
        btnSET3.Enabled = False
        btnSET1.Enabled = False
    Else
        cmbPort.Text = cmbPort.Items.Item(0)
        btnStart.Enabled = True
        btnStop.Enabled = False
        btnLevel.Enabled = False
        btnVolume.Enabled = False
        btnMaketrend.Enabled = False
        btnDatalog.Enabled = True
        btnExit.Enabled = True
        btnSET2.Enabled = False
        btnSET3.Enabled = False
        btnSET1.Enabled = False
        cbBaud.Text = "9600"
        cbBaud.Items.Add(9600)
        cbBaud.Items.Add(19200)
        cbBaud.Items.Add(38400)
        cbBaud.Items.Add(57600)
        cbBaud.Items.Add(115200)
    End If
End Sub

Public databaseName As String = "Pertamina"
Public tableDatalogName As String = "tbl_datalog"
Public tableLimitName As String = "tbl_limit"
Dim statusDatalogTable As Boolean = False
Dim statusLimitTable As Boolean = False
Dim cmdStrDatalog As String = "SELECT COUNT(*)
FROM information_schema.tables WHERE table_schema = '"
& databaseName & "' AND table_name = '" &
tableDatalogName & "'"

```

```

Dim cmdStrLimit As String = "SELECT COUNT(*) FROM
information_schema.tables WHERE table_schema = '" &
databaseName & "' AND table_name = '" & tableLimitName
& "'"

```

```

Private Sub Form2_Load(sender As Object, e As
EventArgs) Handles MyBase.Load

```

```

'cek apakah sudah ada database pertamina
cek_database(databaseName)

```

```

'cek apakah sudah ada table dalam database
cek_table(cmdStrDatalog, "01")
cek_table(cmdStrLimit, "02")

```

```

'buat tableDatalog jika belum ada
If statusDatalogTable = False Then
    createdatalogTable()
End If

```

```

'buat tableLimit jika belum ada
If statusLimitTable = False Then
    createlimitTable()
End If

```

```

serialKonek()
Clock.Start()

```

```

End Sub

```

```

Public connStr As String = "server=localhost;user
id=root;password=;database=" & databaseName & "'"

```

```

Public conn As MySqlConnection = New
MySqlConnection(connStr)

```

```

Private Sub createdatalogTable()

```

```

'create table dataLOG
Dim createSql As String
Try

```

```

    conn = New MySqlConnection(connStr)
    conn.Open()

```

```

    createSql = "CREATE TABLE " &
tableDatalogName & " (no INT(100) NOT NULL
AUTO_INCREMENT,date VARCHAR(15), time VARCHAR(15),

```



```

Vch1 VARCHAR(15),Vch2 VARCHAR(15),Vch3
VARCHAR(15),Lch1 VARCHAR(15),Lch2 VARCHAR(15),Lch3
VARCHAR(15), PRIMARY KEY (no));"
    Dim cmd As New MySqlCommand(createSql,
conn)
        cmd.ExecuteNonQuery()
        cmd.Dispose()
    Catch ex As Exception
        MsgBox(ex.Message)
    Finally
        conn.Close()
    End Try
End Sub

Private Sub createlimitTable()
    'create table dataLOG
    Dim createSql As String
    Try
        conn = New MySqlConnection(connStr)
        conn.Open()
        createSql = "CREATE TABLE " &
tableLimitName & " (no INT(100) NOT NULL
AUTO_INCREMENT,date VARCHAR(15), time VARCHAR(15), PIT
VARCHAR(15),Value VARCHAR(15),SP VARCHAR(15),Deskripsi
VARCHAR(50), PRIMARY KEY (no));"
        Dim cmd As New MySqlCommand(createSql,
conn)
            cmd.ExecuteNonQuery()
            cmd.Dispose()
        Catch ex As Exception
            MsgBox(ex.Message)
        Finally
            conn.Close()
        End Try
    End Sub

    Private Sub cek_table(ByVal command As String,
ByVal ID As String)
        Dim cmd As MySqlCommand = New
MySqlCommand(command, conn)

```

```

conn.Open()

Dim reader As MySqlDataReader =
cmd.ExecuteReader
If (reader.Read() = True) Then
    Dim count As Integer = reader.GetInt32(0)
    If count = 0 Then 'jika tidak ada table
        If ID = "01" Then
            statusDatalogTable = False
        ElseIf ID = "02" Then
            statusLimitTable = False
        End If
    Else 'jika ada table
        If ID = "01" Then
            statusDatalogTable = True
        ElseIf ID = "02" Then
            statusLimitTable = True
        End If
    End If
End If
conn.Close()
End Sub

Private Sub cek_database(ByVal dbName As String)
    Dim conn As MySqlConnection
    Dim strConn As String
    Dim cmd As MySqlCommand

    strConn = "Server =localhost; userid = root;
password =;"
    strConn &= "Database = mysql; pooling=false;"

    conn = New MySqlConnection(strConn)
    cmd = New MySqlCommand("Create Database If Not
exists " & dbName & "", conn)

    conn.Open()

    cmd.ExecuteNonQuery()
    conn.ChangeDatabase(dbName)

```

```

        'create table if not exi
        'Dim sbCmd As New System.Text.StringBuilder
        'sbCmd.Append("Create Table If Not Exists Blog
    ")
        'sbCmd.Append("(ID varchar(36) PRIMARY KEY,
PostDate date, BlogText Text)")
        'cmd = New MySqlCommand(sbCmd.ToString, conn)
        'cmd.ExecuteNonQuery()

        conn.Close()
    End Sub

    Dim statusAplikasi As String = "Stop"
    Private Sub btnStart_Click(sender As Object, e As
EventArgs) Handles btnStart.Click
        SerialPort1.PortName = cmbPort.Text
        'Set SerialPort1 to the selected COM port at startup
        SerialPort1.BaudRate = cbBaud.Text
        'Set Baud rate to the selected value on

        'Other Serial Port Property
        SerialPort1.Parity = IO.Ports.Parity.None
        SerialPort1.StopBits = IO.Ports.StopBits.One
        SerialPort1.DataBits = 8           'Open our
serial port
        SerialPort1.ReadBufferSize = 16
        SerialPort1.Handshake = Handshake.None
        SerialPort1.Encoding =
System.Text.Encoding.Default
        SerialPort1.ReadTimeout = 16
        SerialPort1.Open()

        Timer1.Start()
        statusAplikasi = "Start"

        btnStart.Enabled = False
        btnStop.Enabled = True
        btnLevel.Enabled = True
        btnVolume.Enabled = True

```

```

        btnMaketrend.Enabled = True
        btnDataLog.Enabled = True
        btnExit.Enabled = True
        btnSET1.Enabled = True
        btnSET2.Enabled = True
        btnSET3.Enabled = True
    End Sub

    Private Sub btnStop_Click(sender As Object, e As
EventArgs) Handles btnStop.Click
        If SerialPort1.IsOpen Then
            Timer1.Stop()
            SerialPort1.Close()
            'MsgBox("Disconnected")
        End If
        reset()
    End Sub

    Sub reset()
        btnStart.Enabled = True
        btnStop.Enabled = False
        btnLevel.Enabled = False
        btnVolume.Enabled = False
        btnMaketrend.Enabled = False
        btnDataLog.Enabled = True
        btnExit.Enabled = True
        btnSET2.Enabled = False
        btnSET3.Enabled = False
        btnSET1.Enabled = False

        txtdatalevel1.Text = "0"
        txtdatalevel2.Text = "0"
        txtdatalevel3.Text = "0"
        txtVolume1.Text = "0"
        txtVolume2.Text = "0"
        txtVolume3.Text = "0"

        txtChannel1.Text = "A"
        txtChannel2.Text = "B"
        txtChannel3.Text = "C"
    End Sub

```

```

        progresBar1.Height = 210 - 0
        progresBar2.Height = 210 - 0
        progresBar3.Height = 210 - 0

        statusChanel1 = False
        statusChanel2 = False
        statusChanel3 = False

        statusAplikasi = "Stop"
    End Sub

    'Public Sub DoUpdate(ByVal sender As Object, ByVal
e As System.EventArgs)
    '    'Pada TextBox1.Text sudah dapat menampilkan
data dari SerialPort1
    '    'a = Val(txtPL.Text)
    '    Timer1.Enabled = True
    '    Call proses_fix()
    '    'Try
    '    'engChart.Points.AddXY(data(1), data(2))
    '    'Catch ex As Exception
    '    'Label4.Text = ex.Message
    '    'Finally
    '    'End Try

    '    'Jika membaca boolean disconnect maka
mengirim perintah true ke prosesoff
    '    If disconnect Then
    '        prosesoff = True
    '    End If
    'End Sub

    'Private Sub proses_fix()
    '    strinput = tbdata.Text
    '    Dim panjang_data As Integer
    '    Dim x As Integer
    '    Dim z As Integer
    '    panjang_data = Len(strinput)

```

```

'   Dim i As Integer
'   i = 0
'   z = 0

'   'jika mendeteksi "|" pada string
txt_inputdata dimulai string ke1 dengan panjang data 1
'   For x = 1 To Len(readBuffer$)
'looping untuk data ke dua sampai terakhir pada
txt_inputdata
'       If Mid(readBuffer$, x, 1) = "|" Then
'           z = z + 1
'jika mendeteksi "|" ke dua dimulai ke x dengan
panjang data 1, jadi x sebagai simbol dari "|"
'           Data(i) = Mid(readBuffer$, z, x - z)
'data ke i = integer dari pengambilan data
txt_inputdata dimulai dari z(awalnya 2) sampai ke x -
z
'           i = i + 1
'i incr dari 0 ke satu, satu ke dua dst...
'           z = x
'z di incrm dengan x karena panjang max = x dan awal
data = z maka untuk mendapatkan data = x-z
'       End If
'   Next x
'End Sub

Public levelValue() As String
Sub dataParse(str As String)
    levelValue = str.Split(",")
    'Dim count As Integer
    'For count = 0 To levelValue.Length - 1
    '    MsgBox(levelValue(count))
    'Next
End Sub

Sub kenaLimit(kolam_ As String, value_ As String,
sp_ As String, deskripsi_ As String)
    Dim dt As Date = Date.Today
    conn.Open()

```

```

        Dim createSql As String = "INSERT INTO " &
tableLimitName & "(date, time, PIT, Value, SP,
Deskripsi) VALUES ('" & Date.Now.ToString("d/MM/yyyy")
& "', '" & Date.Now.ToString("hh:mm:ss tt") & "', '" &
kolam_ & "', '" & value_ & "', '" & sp_ & "', '" &
deskripsi_ & "'"")
        Dim cmd As New MySqlCommand(createSql, conn)
        cmd.ExecuteNonQuery()
        cmd.Dispose()
        conn.Close()
    End Sub

```

```

Private Sub Timer1_Tick(sender As Object, e As
EventArgs) Handles Timer1.Tick
    If SerialPort1.IsOpen Then
        Dim dataSerial As String =
SerialPort1.ReadExisting()

        dataParse(dataSerial)

        'MsgBox(levelValue.Length())

        'If dataSerial.Length > 0 And
levelValue(0).Length > 0 And levelValue(1).Length > 0
And levelValue(2).Length > 0 Then
            If dataSerial.Length > 0 And
levelValue.Length() = 3 Then
                If statusChanel1 Then
                    progresBar1.Height = 210 -
Convert.ToInt32(levelValue(0))
                    txtdatalevel1.Text = levelValue(0)
/ 100
                    txtVolume1.Text = (((levelValue(0)
/ 100) * txtPL1.Text) * 1000) / 158.9873 'konversi ke
bbl atau barel
                    txtVolume1.Text =
FormatNumber(txtVolume1.Text, 2)

```

```

        If txtdatalevel1.Text >=
txtHit1.Text Then
            Console.Beep(1750, 50)
            txtChannel1.Text = "A - PENUH"
            kenaLimit("A",
txtdatalevel1.Text, txtHit1.Text, "PENUH")
        ElseIf txtdatalevel1.Text <=
txtHil1.Text Then
            Console.Beep(1750, 50)
            txtChannel1.Text = "A - HABIS"
            kenaLimit("A",
txtdatalevel1.Text, txtHil1.Text, "HABIS")
        Else
            txtChannel1.Text = "A"
        End If
    End If
    If statusChanel2 Then
        progresBar2.Height = 210 -
Convert.ToInt32(levelValue(1))
        txtdatalevel2.Text = levelValue(1)
/ 100
        txtVolume2.Text = (((levelValue(1)
/ 100) * txtPL2.Text) * 1000) / 158.9873 'konversi ke
bbl atau barel
        txtVolume2.Text =
FormatNumber(txtVolume2.Text, 2)

        If txtdatalevel2.Text >=
txtHit2.Text Then
            Console.Beep(1750, 50)
            txtChannel2.Text = "B - PENUH"
            kenaLimit("B",
txtdatalevel2.Text, txtHit2.Text, "PENUH")
        ElseIf txtdatalevel2.Text <=
txtHil2.Text Then
            Console.Beep(1750, 50)
            txtChannel2.Text = "B - HABIS"
            kenaLimit("B",
txtdatalevel2.Text, txtHil2.Text, "HABIS")
        Else

```



```

        txtChannel2.Text = "B"
    End If
End If
If statusChanel3 Then
    progresBar3.Height = 210 -
Convert.ToInt32(levelValue(2))
    txtdatalevel3.Text = levelValue(2)
/ 100
    txtVolume3.Text = (((levelValue(2)
/ 100) * txtPL3.Text) * 1000) / 158.9873 'konversi ke
bbl atau barel
    txtVolume3.Text =
FormatNumber(txtVolume3.Text, 2)

    If txtdatalevel3.Text >=
txtHit3.Text Then
        Console.Beep(1750, 50)
        txtChannel3.Text = "C - PENUH"
        kenaLimit("C",
txtdatalevel3.Text, txtHit3.Text, "PENUH")
    ElseIf txtdatalevel3.Text <=
txtHil3.Text Then
        Console.Beep(1750, 50)
        txtChannel3.Text = "C - HABIS"
        kenaLimit("C",
txtdatalevel3.Text, txtHil3.Text, "HABIS")
    Else
        txtChannel3.Text = "C"
    End If
End If
End If
End If
End Sub

Public trend As String = "Volume"
Private Sub btnVolume_Click(sender As Object, e As
EventArgs) Handles btnVolume.Click
    trend = "Volume"
    Me.Hide()
    Form3.Show()

```

```

        Form3.tmGraph.Start()
    End Sub

    Private Sub btnLevel_Click(sender As Object, e As
EventArgs) Handles btnLevel.Click
        trend = "Level"
        Me.Hide()
        Form3.Show()
        Form3.tmGraph.Start()
    End Sub

    Private Sub btnMaketrend_Click(sender As Object, e
As EventArgs) Handles btnMaketrend.Click
        trend = "Make Trend"
        Me.Hide()
        Form4.Show()
        Form4.tmGraph.Start()
    End Sub

    Private Sub btnDatalog_Click(sender As Object, e
As EventArgs) Handles btnDatalog.Click
        Me.Hide()
        frm_Datalog.Show()
    End Sub

    Public statusChanel1 As Boolean = False
    Public statusChanel2 As Boolean = False
    Public statusChanel3 As Boolean = False
    Private Sub btnSET_Click(sender As Object, e As
EventArgs) Handles btnSET1.Click
        If txtHit1.TextLength = 0 Then
            MsgBox("Hi-T Channel 1 harus diisi !")
            Exit Sub
        End If
        If txtPL1.TextLength = 0 Then
            MsgBox("PxL Channel 1 harus diisi !")
            Exit Sub
        End If
        If txtHil1.TextLength = 0 Then
            MsgBox("Hi-Limit Channel 1 harus diisi !")

```

```

        Exit Sub
    End If

    If statusAplikasi = "Stop" Then
        btnStart.Enabled = True
    End If

    If statusChanel1 Then
        statusChanel1 = False
        progresBar1.Height = 210 - 0
        txtdatalevel1.Text = "0"
        txtVolume1.Text = "0"
        txtChannel1.Text = "A"
    Else
        statusChanel1 = True
    End If
End Sub

Private Sub btnSET2_Click(sender As Object, e As
EventArgs) Handles btnSET2.Click
    If txtHit2.TextLength = 0 Then
        MsgBox("Hi-T Channel 2 harus diisi !")
        Exit Sub
    End If
    If txtPL2.TextLength = 0 Then
        MsgBox("PxL Channel 2 harus diisi !")
        Exit Sub
    End If
    If txtHil2.TextLength = 0 Then
        MsgBox("Hi-Limit Channel 2 harus diisi !")
        Exit Sub
    End If

    If statusAplikasi = "Stop" Then
        btnStart.Enabled = True
    End If

    If statusChanel2 Then
        statusChanel2 = False
        progresBar2.Height = 210 - 0

```

```

        txtdatalevel2.Text = "0"
        txtVolume2.Text = "0"
        txtChannel2.Text = "B"
    Else
        statusChanel2 = True
    End If
End Sub

Private Sub btnSET3_Click(sender As Object, e As
EventArgs) Handles btnSET3.Click
    If txtHit3.TextLength = 0 Then
        MsgBox("Hi-T Channel 3 harus diisi !")
        Exit Sub
    End If
    If txtPL3.TextLength = 0 Then
        MsgBox("PxL Channel 3 harus diisi !")
        Exit Sub
    End If
    If txtHil3.TextLength = 0 Then
        MsgBox("Hi-Limit Channel 3 harus diisi !")
        Exit Sub
    End If

    If statusAplikasi = "Stop" Then
        btnStart.Enabled = True
    End If

    If statusChanel3 Then
        statusChanel3 = False
        progresBar3.Height = 210 - 0
        txtdatalevel3.Text = "0"
        txtVolume3.Text = "0"
        txtChannel3.Text = "C"
    Else
        statusChanel3 = True
    End If
End Sub

'Validasi Anggak

```

```

    Private Sub txtHit1_KeyPress(sender As Object, e
As KeyPressEventArgs) Handles txtHit1.KeyPress
        If Not (e.KeyChar >= "0" And e.KeyChar <= "9"
Or e.KeyChar = vbBack Or e.KeyChar = ".") Then
            e.Handled = True
        End If
    End Sub
    'Validasi Anggak
    Private Sub txtPL1_KeyPress(sender As Object, e As
KeyPressEventArgs) Handles txtPL1.KeyPress
        If Not (e.KeyChar >= "0" And e.KeyChar <= "9"
Or e.KeyChar = vbBack Or e.KeyChar = ".") Then
            e.Handled = True
        End If
    End Sub
    'Validasi Anggak
    Private Sub txtHil1_KeyPress(sender As Object, e
As KeyPressEventArgs) Handles txtHil1.KeyPress
        If Not (e.KeyChar >= "0" And e.KeyChar <= "9"
Or e.KeyChar = vbBack Or e.KeyChar = ".") Then
            e.Handled = True
        End If
    End Sub
    'Validasi Anggak
    Private Sub txtHit2_KeyPress(sender As Object, e
As KeyPressEventArgs) Handles txtHit2.KeyPress
        If Not (e.KeyChar >= "0" And e.KeyChar <= "9"
Or e.KeyChar = vbBack Or e.KeyChar = ".") Then
            e.Handled = True
        End If
    End Sub
    'Validasi Anggak
    Private Sub txtPL2_KeyPress(sender As Object, e As
KeyPressEventArgs) Handles txtPL2.KeyPress
        If Not (e.KeyChar >= "0" And e.KeyChar <= "9"
Or e.KeyChar = vbBack Or e.KeyChar = ".") Then
            e.Handled = True
        End If
    End Sub
    'Validasi Anggak

```

```

Private Sub txtHil2_KeyPress(sender As Object, e
As KeyPressEventArgs) Handles txtHil2.KeyPress
    If Not (e.KeyChar >= "0" And e.KeyChar <= "9"
Or e.KeyChar = vbBack Or e.KeyChar = ".") Then
        e.Handled = True
    End If
End Sub
'Validasi Anggak
Private Sub txtHit3_KeyPress(sender As Object, e
As KeyPressEventArgs) Handles txtHit3.KeyPress
    If Not (e.KeyChar >= "0" And e.KeyChar <= "9"
Or e.KeyChar = vbBack Or e.KeyChar = ".") Then
        e.Handled = True
    End If
End Sub
'Validasi Anggak
Private Sub txtPL3_KeyPress(sender As Object, e As
KeyPressEventArgs) Handles txtPL3.KeyPress
    If Not (e.KeyChar >= "0" And e.KeyChar <= "9"
Or e.KeyChar = vbBack Or e.KeyChar = ".") Then
        e.Handled = True
    End If
End Sub
'Validasi Anggak
Private Sub txtHil3_KeyPress(sender As Object, e
As KeyPressEventArgs) Handles txtHil3.KeyPress
    If Not (e.KeyChar >= "0" And e.KeyChar <= "9"
Or e.KeyChar = vbBack Or e.KeyChar = ".") Then
        e.Handled = True
    End If
End Sub

Private Sub Form2_FormClosed(sender As Object, e
As FormClosedEventArgs) Handles MyBase.FormClosed
    Dim Ingin As MsgBoxResult
    Ingin = MsgBox("Apakah Anda akan keluar?",
MsgBoxStyle.YesNo, "Keluar")
    If Ingin = MsgBoxResult.Yes Then
        If SerialPort1.IsOpen Then
            SerialPort1.Close()

```

```

        End If
        reset()
        Clock.Stop()
        Form1.Show()
        Me.Hide()
    ElseIf Ingin = MsgBoxResult.No Then
        Exit Sub
    End If
End Sub

Private Sub Clock_Tick(sender As Object, e As
EventArgs) Handles Clock.Tick
    Dim dt As Date = Date.Today
    Dim hari As String = ""
    If dt.ToString("dddd") = "Sunday" Then
        hari = "MINGGU"
    ElseIf dt.ToString("dddd") = "Monday" Then
        hari = "SENIN"
    ElseIf dt.ToString("dddd") = "Tuesday" Then
        hari = "SELASA"
    ElseIf dt.ToString("dddd") = "Wednesday" Then
        hari = "RABU"
    ElseIf dt.ToString("dddd") = "Thursday" Then
        hari = "KAMIS"
    ElseIf dt.ToString("dddd") = "Friday" Then
        hari = "JUM'AT"
    ElseIf dt.ToString("dddd") = "Saturday" Then
        hari = "SABTU"
    End If
    txt_clock.Text = hari + ", " +
Date.Now.ToString("d/MM/yyyy hh:mm:ss tt")

    Dim allValue(6) As String
    If Timer1.Enabled = True And
(txtdatalevel1.Text > 0 Or txtdatalevel2.Text > 0 Or
txtdatalevel3.Text > 0) Then
        If statusChanel1 Then
            allValue(0) = txtVolume1.Text
            allValue(3) = txtdatalevel1.Text
        Else

```

```

        allValue(0) = "-"
        allValue(3) = "-"
    End If

    If statusChanel2 Then
        allValue(1) = txtVolume2.Text
        allValue(4) = txtdatalevel2.Text
    Else
        allValue(1) = "-"
        allValue(4) = "-"
    End If

    If statusChanel3 Then
        allValue(2) = txtVolume3.Text
        allValue(5) = txtdatalevel3.Text
    Else
        allValue(2) = "-"
        allValue(5) = "-"
    End If

    conn.Open()
    Dim createSql As String = "INSERT INTO " &
tableDatalogName & "(date, time, Vch1, Vch2, Vch3,
Lch1, Lch2, Lch3) VALUES ('" &
Date.Now.ToString("d/MM/yyyy") & "', '" &
Date.Now.ToString("hh:mm:ss tt") & "', '" & allValue(0)
& "', '" & allValue(1) & "', '" & allValue(2) & "', '" &
allValue(3) & "', '" & allValue(4) & "', '" & allValue(5)
& "'"")"

    Dim cmd As New MySqlCommand(createSql,
conn)

    cmd.ExecuteNonQuery()
    cmd.Dispose()
    conn.Close()
End If
End Sub

End Class

```



#### A-4. Listing Program Visual Basic Form Grafik

```
Imports
System.Windows.Forms.DataVisualization.Charting

Public Class Form3

    Dim RXArray(2047) As Char
    Dim RXCnt As Integer
    Dim engChart As New
System.Windows.Forms.DataVisualization.Charting.Series
    Dim data(5) As String

    Private Sub Form3_Load(sender As Object, e As
EventArgs) Handles MyBase.Load
        'tmGraph.Start()
        Count = 0
    End Sub

    Dim Count As Long
    Private Sub tmGraph_Tick(sender As Object, e As
EventArgs) Handles tmGraph.Tick
        Chart1.Titles(2).Text = "Time (" & Count &
" s)"
        If Form2.trend = "Volume" Then
            Chart1.Titles(0).Text = "Value Of
Volume"
            Chart1.Series(0).Name = "Volume CH_1"
            Chart1.Series(1).Name = "Volume CH_2"
            Chart1.Series(2).Name = "Volume CH_3"
            If Form2.levelValue.Length >= 3 Then
                If Form2.statusChanel1 Then
                    Chart1.Series(0).Points.AddY(Form2.txtVolume1.Text)
                    End If
                    If Form2.statusChanel2 Then
                        Chart1.Series(1).Points.AddY(Form2.txtVolume2.Text)
                        End If
                End If
            End If
        End If
    End Sub
End Class
```

```

        If Form2.statusChanel3 Then

Chart1.Series(2).Points.AddY(Form2.txtVolume3.Text)
        End If
        Count += 1
        End If
    ElseIf Form2.trend = "Level" Then
Chart1.Titles(0).Text = "Value Of
Level"
        Chart1.Series(0).Name = "Level CH_1"
        Chart1.Series(1).Name = "Level CH_2"
        Chart1.Series(2).Name = "Level CH_3"
        If Form2.levelValue.Length >= 3 Then
            If Form2.statusChanel1 Then

Chart1.Series(0).Points.AddY(Form2.levelValue(0) /
100)
                End If
                If Form2.statusChanel2 Then

Chart1.Series(1).Points.AddY(Form2.levelValue(1) /
100)
                    End If
                    If Form2.statusChanel3 Then

Chart1.Series(2).Points.AddY(Form2.levelValue(2) /
100)
                        End If
                        Count += 1
                        End If
        ElseIf Form2.trend = "Make Trend" Then
Chart1.Titles(0).Text = "Trend Volume &
Level"
        Chart1.Series(0).Name = "Trend Volume"
        Chart1.Series(1).Name = "Trend Level"
        Chart1.Series(2).Name = "-"
        Dim totalChanelAktif As Integer = 0
        Dim volumeTrend As Double = 0
        Dim levelTrend As Double = 0
        If Form2.levelValue.Length >= 3 Then

```

```

                If Form2.statusChanel1 Then
                    totalChanelAktif += 1
                    volumeTrend +=
Form2.txtVolume1.Text
                    levelTrend +=
Form2.txtdatalevel1.Text
                End If
                If Form2.statusChanel2 Then
                    totalChanelAktif += 1
                    volumeTrend +=
Form2.txtVolume2.Text
                    levelTrend +=
Form2.txtdatalevel2.Text
                End If
                If Form2.statusChanel3 Then
                    totalChanelAktif += 1
                    volumeTrend +=
Form2.txtVolume3.Text
                    levelTrend +=
Form2.txtdatalevel3.Text
                End If

                If totalChanelAktif > 0 Then

Chart1.Series(0).Points.AddY(volumeTrend /
totalChanelAktif)

Chart1.Series(1).Points.AddY(levelTrend /
totalChanelAktif)
                    Count += 1
                End If
            End If
        End If

        If Chart1.Series(0).Points.Count > 20 Or
Chart1.Series(1).Points.Count > 20 Or
Chart1.Series(2).Points.Count > 20 Then
            If Form2.trend = "Make Trend" Then
                Chart1.Series(0).Points.RemoveAt(0)
                Chart1.Series(1).Points.RemoveAt(0)

```

```

        Else
            If Form2.statusChanel1 Then

Chart1.Series(0).Points.RemoveAt(0)
                End If
            If Form2.statusChanel2 Then

Chart1.Series(1).Points.RemoveAt(0)
                End If
            If Form2.statusChanel3 Then

Chart1.Series(2).Points.RemoveAt(0)
                End If
            End If
        End If
    End Sub

    Private Sub Form3_FormClosed(sender As Object,
e As FormClosedEventArgs) Handles MyBase.FormClosed
        tmGraph.Stop()
        Count = 0
        Chart1.Series(0).Points.Clear()
        Chart1.Series(1).Points.Clear()
        Chart1.Series(2).Points.Clear()
        Form2.Show()
        Me.Hide()
    End Sub

    Private Sub Chart1_Click(sender As Object, e As
EventArgs) Handles Chart1.Click

    End Sub
End Class

```

#### **A-5. Listing Program Visual Basic Form Make Trend**

```

Imports
System.Windows.Forms.DataVisualization.Charting

Public Class Form4

```

```

Dim RXArray(2047) As Char
Dim RXCnt As Integer
Dim engChart As New
System.Windows.Forms.DataVisualization.Charting.Series
Dim data(5) As String

Private Sub Form4_Load(sender As Object, e As
EventArgs) Handles MyBase.Load
    'tmGraph.Start()
    Count = 0
    Chart2.Series(0).Color = Color.Orange
End Sub

Dim Count As Long
Private Sub tmGraph_Tick(sender As Object, e As
EventArgs) Handles tmGraph.Tick
    Chart1.Titles(2).Text = "Time (" & Count & "
s)"
    Chart2.Titles(2).Text = "Time (" & Count & "
s)"
    If Form2.trend = "Make Trend" Then
        Chart1.Titles(0).Text = "Trend Volume &
Level"
        Chart1.Series(0).Name = "Trend Volume"
        Chart2.Series(0).Name = "Trend Level"
        Dim totalChanelAktif As Integer = 0
        Dim volumeTrend As Double = 0
        Dim levelTrend As Double = 0
        If Form2.levelValue.Length >= 3 Then
            If Form2.statusChanel1 Then
                totalChanelAktif += 1
                volumeTrend +=
Form2.txtVolume1.Text
                levelTrend +=
Form2.txtdatalevel1.Text
            End If
            If Form2.statusChanel2 Then
                totalChanelAktif += 1

```

```

        volumeTrend +=
Form2.txtVolume2.Text
        levelTrend +=
Form2.txtdatalevel2.Text
    End If
    If Form2.statusChanel3 Then
        totalChanelAktif += 1
        volumeTrend +=
Form2.txtVolume3.Text
        levelTrend +=
Form2.txtdatalevel3.Text
    End If

    If totalChanelAktif > 0 Then

Chart1.Series(0).Points.AddY(volumeTrend /
totalChanelAktif)

Chart2.Series(0).Points.AddY(levelTrend /
totalChanelAktif)
        Count += 1
    End If
    End If
End If

If Chart1.Series(0).Points.Count > 20 Then
    If Form2.trend = "Make Trend" Then
        Chart1.Series(0).Points.RemoveAt(0)
    End If
End If

If Chart2.Series(0).Points.Count > 20 Then
    If Form2.trend = "Make Trend" Then
        Chart2.Series(0).Points.RemoveAt(0)
    End If
End If
End Sub

Private Sub Form4_FormClosed(sender As Object, e
As FormClosedEventArgs) Handles MyBase.FormClosed

```

```

        tmGraph.Stop()
        Count = 0
        Chart1.Series(0).Points.Clear()
        Chart2.Series(0).Points.Clear()
        Form2.Show()
        Me.Hide()
    End Sub
End Class

```

#### **A-6. Listing Program Visual Basic Form Datalog**

```

Imports MySql.Data.MySqlClient
Imports Microsoft.Office.Interop

Public Class frm_Datalog

    Private Sub frm_Datalog_Load(sender As Object,
e As EventArgs) Handles MyBase.Load
        open_datalog("1")
        atur_Tabel_datalog()
        open_limit()
        atur_Tabel_limit()

        cmb_chanel.Text = "1"
    End Sub

    Private Sub open_datalog(chanel As String)
        Dim pieceofQuery As String = ""
        Dim conditionofQuery As String = ""
        If chanel = 1 Then
            pieceofQuery = "Vch1, Lch1"
            conditionofQuery = "WHERE Vch1 != '-'"
OR Lch1 != '-'"
        ElseIf chanel = 2 Then
            pieceofQuery = "Vch2, Lch2"
            conditionofQuery = "WHERE Vch2 != '-'"
OR Lch2 != '-'"
        ElseIf chanel = 3 Then
            pieceofQuery = "Vch3, Lch3"

```

```

        conditionofQuery = "WHERE Vch3 != '-'"
OR Lch3 != '-'"
    End If
    Form2.conn.Open()
    Dim mDA As New MySqlDataAdapter("SELECT no,
date, time, " & pieceofQuery & " FROM " &
Form2.tableDatalogName & " " & conditionofQuery &
"", Form2.conn)
    Dim datalog As New DataTable
    mDA.Fill(datalog)
    dgridDatalog.DataSource = datalog
    Form2.conn.Close()

    'untuk mengganti id pada database dengan
    nomor yang terurut, karena ada kemungkinan id pada
    database tidak terurut
    For x As Integer = 0 To
dgridDatalog.RowCount - 1 'dikurangi dua karena row
datagrid dimulai dari 0, dan ada lebih 1 row
        dgridDatalog.Rows(x).Cells(0).Value = x
+ 1 'ditambah 1 agar nomor urut dimulai dari 1
    Next
End Sub

Private Sub open_limit()
    Form2.conn.Open()
    Dim mDA As New MySqlDataAdapter("SELECT no,
date, time, PIT, Value, SP, Deskripsi FROM " &
Form2.tableLimitName & "", Form2.conn)
    Dim datalog As New DataTable
    mDA.Fill(datalog)
    dgridLimit.DataSource = datalog
    Form2.conn.Close()

    'untuk mengganti id pada database dengan
    nomor yang terurut, karena ada kemungkinan id pada
    database tidak terurut
    For x As Integer = 0 To
dgridDatalog.RowCount - 1 'dikurangi dua karena row
datagrid dimulai dari 0, dan ada lebih 1 row

```



```

        dgridDatalog.Rows(x).Cells(0).Value = x
+ 1 'ditambah 1 agar nomor urut dimulai dari 1
    Next
End Sub

```

```

Sub atur_Tabel_datalog()
    Try
        With dgridDatalog
            .Columns(0).Width = 36
            .Columns(0).HeaderText = "No."
            .Columns(1).Width = 100
            .Columns(1).HeaderText = "Date"
            .Columns(2).Width = 100
            .Columns(2).HeaderText = "Time"
            .Columns(3).Width = 100
            .Columns(3).HeaderText = "Volume"
            .Columns(4).Width = 100
            .Columns(4).HeaderText = "Level"
        End With
    Catch ex As Exception
    End Try
End Sub

```

```

Sub atur_Tabel_limit()
    Try
        With dgridLimit
            .Columns(0).Width = 36
            .Columns(0).HeaderText = "No."
            .Columns(1).Width = 100
            .Columns(1).HeaderText = "Date"
            .Columns(2).Width = 100
            .Columns(2).HeaderText = "Time"
            .Columns(3).Width = 50
            .Columns(3).HeaderText = "PIT"
            .Columns(4).Width = 50
            .Columns(4).HeaderText = "Value"
            .Columns(5).Width = 50
            .Columns(5).HeaderText = "SP"
            .Columns(6).Width = 100
        End With
    End Try
End Sub

```

```

        .Columns(6).HeaderText =
"Deskripsi"
        End With
        Catch ex As Exception
        End Try
    End Sub

    Private Sub frmDatalog_FormClosed(sender As
Object, e As FormClosedEventArgs) Handles
MyBase.FormClosed
        Form2.Show()
        Me.Hide()
    End Sub

    Private Sub
cmb_chanel_SelectedIndexChanged(sender As Object, e
As EventArgs) Handles
cmb_chanel.SelectedIndexChanged
        open_datalog(cmb_chanel.Text)
        atur_Tabel_datalog()
    End Sub

    Private Sub btn_resetDatalog_Click(sender As
Object, e As EventArgs) Handles
btn_resetDatalog.Click
        Dim pesan As Integer
        pesan = MsgBox("Anda yakin akan menghapus
semua data sensor ?", vbQuestion + vbYesNo)
        If pesan = vbYes Then
            Form2.conn.Open()
            'delete semua row pada table dan
mengembalikan auto_increment menjadi 1
            Dim mDA As New
MySqlDataAdapter("truncate " &
Form2.tableDatalogName & "", Form2.conn)
            Dim dt As New DataTable
            mDA.Fill(dt)
            Form2.conn.Close()

            open_datalog("1")

```

```

        atur_Tabel_datalog()
        cmb_chanel.Text = "1"
    End If
End Sub

Private Sub btn_ResetLimit_Click(sender As
Object, e As EventArgs) Handles
btn_ResetLimit.Click
    Dim pesan As Integer
    pesan = MsgBox("Anda yakin akan menghapus
semua data alarm record ?", vbQuestion + vbYesNo)
    If pesan = vbYes Then
        Form2.conn.Open()
        'delete semua row pada table dan
mengembalikan auto_increment menjadi 1
        Dim mDA As New
MySqlDataAdapter("truncate " & Form2.tableLimitName
& "", Form2.conn)
        Dim dt As New DataTable
        mDA.Fill(dt)
        Form2.conn.Close()

        open_limit()
        atur_Tabel_limit()
    End If
End Sub

Private Sub btnRefresh_Click(sender As Object,
e As EventArgs) Handles btnRefresh.Click
    open_datalog(cmb_chanel.Text)
    atur_Tabel_datalog()
    open_limit()
    atur_Tabel_limit()
End Sub

Private Sub btnSaveDatalog_Click(sender As
Object, e As EventArgs) Handles
btnSaveDatalog.Click
    Dim saveFileDialog1 As New SaveFileDialog

```

```

        saveFileDialog1.Filter = "Excel
Files(.xlsx)|*"
        saveFileDialog1.Title = "Save an Excel
File"
        saveFileDialog1.ShowDialog()
        If saveFileDialog1.FileName <> "" Then
            Dim xlApp As Excel.Application
            Dim xlWorkBook As Excel.Workbook
            Dim xlWorkSheet As Excel.Worksheet
            Dim misValue As Object =
System.Reflection.Missing.Value

            Dim i As Int16, j As Int16

            xlApp = New Excel.Application
            xlWorkBook =
xlApp.Workbooks.Add(misValue)
            xlWorkSheet =
xlWorkBook.Sheets("sheet1")

            For x = 0 To dgridDatalog.ColumnCount -
1
                xlWorkSheet.Cells(1, x + 1) = _
dgridDatalog.Columns(x).HeaderText
            Next

            For i = 0 To dgridDatalog.RowCount - 1
                For j = 0 To
dgridDatalog.ColumnCount - 1
                    xlWorkSheet.Cells(i + 2, j + 1)
= dgridDatalog(j, i).Value.ToString()
                Next
            Next

            Dim chartRange As Excel.Range
            chartRange = xlWorkSheet.Range("a1",
"e1")

            chartRange.Font.Bold = True
            chartRange = xlWorkSheet.Range("a1",
"e1")

```

```

        chartRange.Interior.Color =
System.Drawing.ColorTranslator.ToOle(System.Drawing.
Color.Navy)
        chartRange.Font.Color =
System.Drawing.ColorTranslator.ToOle(System.Drawing.
Color.White)

```

```

        chartRange = xlWorkSheet.Range("a1")
        chartRange.Cells.ColumnWidth = 4
        chartRange = xlWorkSheet.Range("b1")
        chartRange.Cells.ColumnWidth = 20
        chartRange = xlWorkSheet.Range("c1")
        chartRange.Cells.ColumnWidth = 20
        chartRange = xlWorkSheet.Range("d1")
        chartRange.Cells.ColumnWidth = 20
        chartRange = xlWorkSheet.Range("e1")
        chartRange.Cells.ColumnWidth = 20

```

```

xlWorkBook.SaveAs(saveFileDialog1.FileName + ".xls",
Excel.XlFileFormat.xlWorkbookNormal, misValue,
misValue, misValue, misValue, _
        Excel.XlSaveAsAccessMode.xlExclusive,
misValue, misValue, misValue, misValue, misValue)
        xlWorkBook.Close(True, misValue,
misValue)
        xlApp.Quit()

        releaseObject(xlWorkSheet)
        releaseObject(xlWorkBook)
        releaseObject(xlApp)

        Console.Beep(2500, 100)
        MessageBox.Show("File tersimpan !")
    End If
End Sub

```

```

Private Sub btnSaveLimit_Click(sender As Object,
e As EventArgs) Handles btnSaveLimit.Click
    Dim saveFileDialog1 As New SaveFileDialog

```

```

        saveFileDialog1.Filter = "Excel
Files(.xlsx)|*"
        saveFileDialog1.Title = "Save an Excel
File"
        saveFileDialog1.ShowDialog()
        If saveFileDialog1.FileName <> "" Then
            Dim xlApp As Excel.Application
            Dim xlWorkBook As Excel.Workbook
            Dim xlWorkSheet As Excel.Worksheet
            Dim misValue As Object =
System.Reflection.Missing.Value

            Dim i As Int16, j As Int16

            xlApp = New Excel.Application
            xlWorkBook =
xlApp.Workbooks.Add(misValue)
            xlWorkSheet =
xlWorkBook.Sheets("sheet1")

            For x = 0 To dgridLimit.ColumnCount - 1
                xlWorkSheet.Cells(1, x + 1) = _
dgridLimit.Columns(x).HeaderText
            Next

            For i = 0 To dgridLimit.RowCount - 1
                For j = 0 To dgridLimit.ColumnCount
- 1
                    xlWorkSheet.Cells(i + 2, j + 1)
= dgridLimit(j, i).Value.ToString()
                Next
            Next

            Dim chartRange As Excel.Range
            chartRange = xlWorkSheet.Range("a1",
"g1")

            chartRange.Font.Bold = True
            chartRange = xlWorkSheet.Range("a1",
"g1")

```

```

        chartRange.Interior.Color =
System.Drawing.ColorTranslator.ToOle(System.Drawing.
Color.Navy)
        chartRange.Font.Color =
System.Drawing.ColorTranslator.ToOle(System.Drawing.
Color.White)

```

```

        chartRange = xlWorkSheet.Range("a1")
        chartRange.Cells.ColumnWidth = 4
        chartRange = xlWorkSheet.Range("b1")
        chartRange.Cells.ColumnWidth = 20
        chartRange = xlWorkSheet.Range("c1")
        chartRange.Cells.ColumnWidth = 20
        chartRange = xlWorkSheet.Range("d1")
        chartRange.Cells.ColumnWidth = 20
        chartRange = xlWorkSheet.Range("e1")
        chartRange.Cells.ColumnWidth = 20
        chartRange = xlWorkSheet.Range("f1")
        chartRange.Cells.ColumnWidth = 20
        chartRange = xlWorkSheet.Range("g1")
        chartRange.Cells.ColumnWidth = 20

```

```

xlWorkBook.SaveAs(saveFileDialog1.FileName + ".xls",
Excel.XlFileFormat.xlWorkbookNormal, misValue,
misValue, misValue, misValue, _
        Excel.XlSaveAsAccessMode.xlExclusive,
misValue, misValue, misValue, misValue, misValue)
        xlWorkBook.Close(True, misValue,
misValue)

```

```

        xlApp.Quit()

```

```

        releaseObject(xlWorkSheet)
        releaseObject(xlWorkBook)
        releaseObject(xlApp)

```

```

        Console.Beep(2500, 100)
        MessageBox.Show("File tersimpan !")

```

```

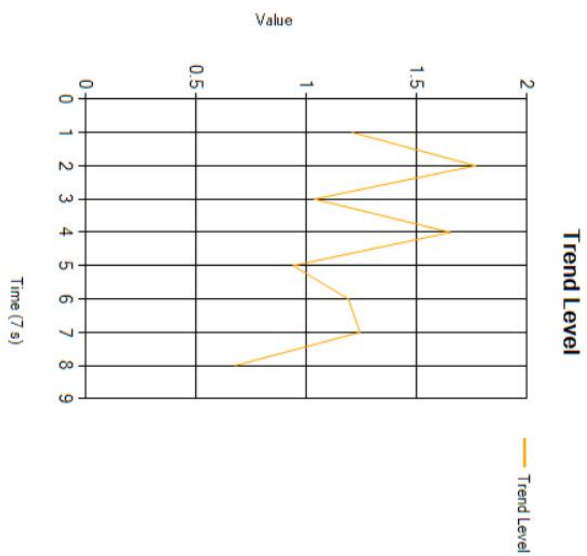
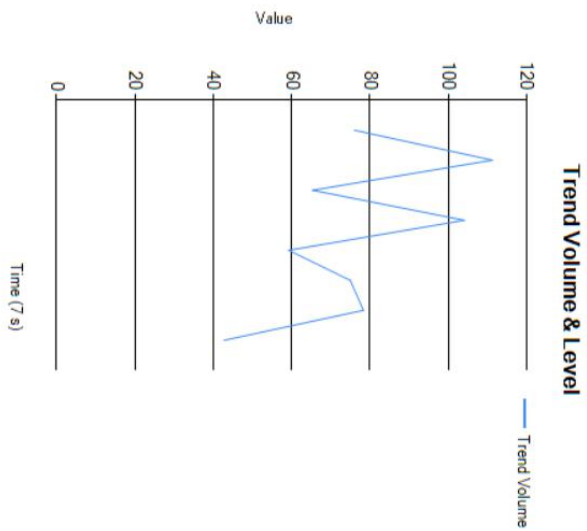
    End If
End Sub

```

```
Private Sub releaseObject(ByVal obj As Object)
    Try
        System.Runtime.InteropServices.Marshal.ReleaseComOb
        ject(obj)
        obj = Nothing
    Catch ex As Exception
        obj = Nothing
        MessageBox.Show("Exception Occured
while releasing object " + ex.ToString())
    Finally
        GC.Collect()
    End Try
End Sub
End Class
```



## B-1. Tampilan Grafik HMI



## C-1. Datasheet



### Features

- Bushing mount
  - Optional AR pin feature
  - Plastic or metal shaft and bushings
  - Wirewound
  - Solder lugs or PC pins
  - Sealable (Full body seal)
  - Designed for use in HMI applications
- RoHS compliant\*

**BOURNS®**

### 3590 - Precision Potentiometer

#### Electrical Characteristics<sup>1</sup>

Standard Resistance Range	200 to 100 K ohms
Total Resistance Tolerance	±3 %
Independent Linearity	±0.25 %
Effective Electrical Angle	3000 ° ±10 °, -0 ° <sup>2</sup>
Absolute Minimum Resistance	1 ohm or 0.1 % maximum (whichever is greater)
Noise	100 ohms ENR maximum
Dielectric Withstanding Voltage (MIL-STD-202, Method 301)	
Sea Level	1,500 VAC minimum
Power Rating (Voltage Limited By Power Dissipation or 450 VAC, Whichever Is Less)	
+40 °C	2 watts
+125 °C	0 watt
Insulation Resistance (500 VDC)	1,000 megohms minimum
Resolution	See recommended part numbers

#### Environmental Characteristics<sup>1</sup>

Operating Temperature Range	-40 °C to +125 °C
Storage Temperature Range	-55 °C to +125 °C
Temperature Coefficient Over Storage Temperature Range <sup>2</sup>	±50 ppm/°C maximum/unit
Vibration	15 G
Wiper Bounce	0.1 millisecond maximum
Shock	50 G
Wiper Bounce	0.1 millisecond maximum
Load Life	1,000 hours, 2 watts
Total Resistance Shift	±2 % maximum
Rotational Life (No Load)	1,000,000 shaft revolutions
Total Resistance Shift	±5 % maximum
Moisture Resistance (MIL-STD-202, Method 103, Condition B)	
Total Resistance Shift	±2 % maximum
IP Rating	
Sealed Versions (-3, -4, -7, and -8)	IP 65
Unsealed Versions (-1, -2, -5, and -9)	IP 40
Moisture Sensitivity Level	1
ESD Classification (HBM)	N/A

#### Mechanical Characteristics<sup>1</sup>

Stip Strength	45 N-cm (64 oz.-in.) minimum
Mechanical Angle	3000 ° ±10 °, -0 ° <sup>2</sup>
Torque	
Starting & Running	0.85 N-cm (1.2 oz.-in.) maximum (unsealed) 1.41 N-cm (2.0 oz.-in.) maximum (sealed)
Mounting	55-80 N-cm (5-7 lb.-in.) (plastic) 90-113 N-cm (8-10 lb.-in.) (metal)
Shaft Runout	0.13 mm (0.005 in.) T.I.R.
Lateral Runout	0.20 mm (0.008 in.) T.I.R.
Shaft End Play	0.25 mm (0.010 in.) T.I.R.
Shaft Radial Play	0.13 mm (0.005 in.) T.I.R.
Pilot Diameter Runout	0.08 mm (0.003 in.) T.I.R.
Backlash	1.0 ° maximum
Weight	Approximately 19 G
Terminals	Solder lugs or PC pins
Soldering Condition	
Manual Soldering	96.5Sn/3.0Ag/0.5Cu solid wire or no-clean rosin cored wire; 370 °C (700 °F) max. for 3 seconds
Wave Soldering	96.5Sn/3.0Ag/0.5Cu solder with no-clean flux; 260 °C (500 °F) max. for 5 seconds
Wash processes	Not recommended
Marking	Manufacturer's name and part number, resistance value and tolerance, linearity tolerance, wiring diagram, and date code.
Ganging (Multiple Section Potentiometers)	1 cup maximum
Hardware	One lockwasher and one mounting nut is shipped with each potentiometer.

NOTE: For anti-rotation pin add 91 after configuration dash number. Example: -2 becomes -291 to add AR pin.

<sup>1</sup> At room ambient; ±25 °C nominal and 50 % relative humidity nominal, except as noted.

<sup>2</sup> Consult manufacturer for complete specification details for resistances below 1k ohms.

\*RoHS Directive 2002/95/EC Jan. 27, 2003 including annex and RoHS Recast 2011/65/EU June 8, 2011.

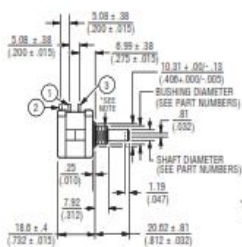
Specifications are subject to change without notice.

Users should verify actual device performance in their specific applications.

The products described herein and this document are subject to specific disclaimers as set forth on the last page of this document, and at [www.bourns.com/legal/disclaimer.pdf](http://www.bourns.com/legal/disclaimer.pdf).

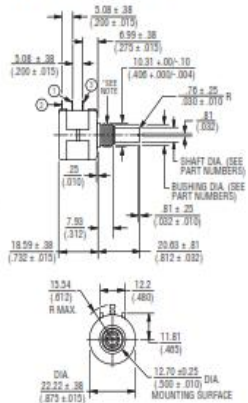
## Product Dimensions

## -1, -3, -5, -7 Configurations

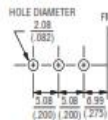


\*BUSHING THREAD SPECIFICATIONS:  
 -1, -2, -3, -4...3/16-32 UNF  
 -5, -6, -7, -8...M9 x 0.75-7H

## -2, -4, -6, -8 Configurations



## Recommended PCB Layout



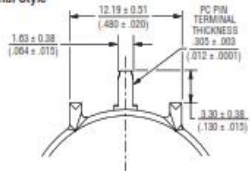
## Schematic



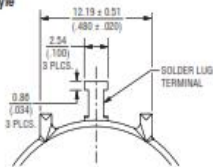
TOLERANCES: EXCEPT WHERE NOTED  
 DECIMALS: XX ± .016 XXX ± .021  
 FRACTIONS: ±1/64  
 DIMENSIONS: MM (IN)

## Terminal Styles

## "P" Terminal Style



## "S" Terminal Style



Specifications are subject to change without notice.

Users should verify actual device performance in their specific applications.

The products described herein and this document are subject to specific disclaimers as set forth on the last page of this document, and at [www.bourns.com/legal/disclaimer.pdf](http://www.bourns.com/legal/disclaimer.pdf)

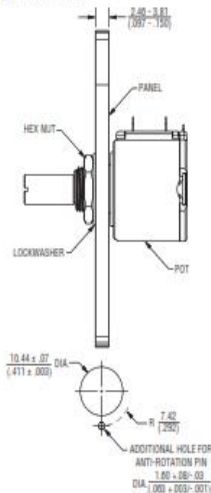
## Recommended Part Numbers

Printed Circuit	Solder Lug	Solder Lug	Resistance (Ω)	Resolution (%)
3590P-2-102L	3590S-2-102L	3090S-1-102L	1,000	.029
3590P-2-202L	3590S-2-202L	3090S-1-202L	2,000	.023
3590P-2-502L	3590S-2-502L	3090S-1-502L	5,000	.023
3590P-2-103L	3590S-2-103L	3090S-1-103L	10,000	.020
3590P-2-203L	3590S-2-203L	3090S-1-203L	20,000	.019
3590P-2-503L	3590S-2-503L	3090S-1-503L	50,000	.013
3590P-2-104L	3590S-2-104L	3090S-1-104L	100,000	.009

SOLENOID LEADINGS ARE IN STOCK AND READILY AVAILABLE THROUGH DISTRIBUTION. FOR OTHER OPTIONS CONSULT FACTORY.  
 FORM IDENTIFIER:  
 L = COMPLIANT

## Panel Thickness Dimensions

(For Bushing Mount Only)



Anti-rotation pin hole is shown at six o'clock position for reference only. The actual location is determined by the customer's application. Refer to the front view of the potentiometer to see the location of the optional A/R pin.

Panel thickness and hole diameters are recommended for best fit. However, customers may adjust the dimensions to suit their specific application.

## Shaft &amp; Bushing Configurations

(Bushing - Dxl, Shaft - D):

- (-1) Plastic Bushing (3/8" x 5/16") and Shaft (.2480 + .001, -.002)
- (-2) Metal Bushing (3/8" x 5/16") and Shaft (.2497 + .0000, -.0009)
- (-3) Sealed, Plastic Bushing (3/8" x 5/16") and Shaft (.2480 + .001, -.002)
- (-4) Sealed, Metal Bushing (3/8" x 5/16") and Shaft (.2497 + .0000, -.0009)
- (-5) Metric, Plastic Bushing (9 mm x 7.94 mm) and Shaft (6 mm + 0, -.076 mm)
- (-6) Metric, Metal Bushing (9 mm x 7.94 mm) and Shaft (6 mm + 0, -.023 mm)
- (-7) Metric, Sealed, Plastic Bushing (9 mm x 7.94 mm) and Shaft (6 mm + 0, -.076 mm)
- (-8) Metric, Sealed, Metal Bushing (9 mm x 7.94 mm) and Shaft (6 mm + 0, -.023 mm)

## REV. 11/17

Specifications are subject to change without notice.  
 Users should verify actual device performance in their specific applications.  
 The products described herein and this document are subject to specific disclaimers as set forth on the last page of this document, and at [www.bourns.com/html/dfs/index.asp](http://www.bourns.com/html/dfs/index.asp)

## Legal Disclaimer Notice

**BOURNS®**

This legal disclaimer applies to purchasers and users of Bourns® products manufactured by or on behalf of Bourns, Inc. and its affiliates (collectively, "Bourns").

Unless otherwise expressly indicated in writing, Bourns® products and data sheets relating thereto are subject to change without notice. Users should check for and obtain the latest relevant information before placing orders and should verify that such information is current and complete.

The characteristics and parameters of a Bourns® product set forth in its data sheet are based on laboratory conditions, and statements regarding the suitability of products for certain types of applications are based on Bourns' knowledge of typical requirements in generic applications. The characteristics and parameters of a Bourns® product in a user application may vary from the data sheet characteristics and parameters due to a combination of the Bourns® product with other components in the user's application or due to the environment of the user application itself. Such characteristics and parameters also can and do vary in different applications and actual performance may vary over time. Users should always verify actual performance of the Bourns® product in their specific devices and applications, and make their own independent judgments about how much additional test margin to design in to compensate for differences between laboratory and real world conditions.

Unless Bourns has explicitly designated an individual Bourns® product as meeting the requirements of a particular industry standard (e.g., ISO/TS 16949) or a particular qualification (e.g., UL listed or recognized), Bourns is not responsible for any failure of an individual Bourns® product to meet requirements of such industry standard or such particular qualification. Users of Bourns® products are responsible for ensuring compliance with safety-related requirements and standards applicable to their applications.

Bourns® products are not recommended, authorized or intended for use in nuclear, lifesaving, life-critical or life-sustaining applications, nor in any other applications where failure or malfunction may result in personal injury, death, or severe property or environmental damage. Unless expressly and specifically approved in writing by two authorized Bourns representatives on a case-by-case basis, use of any Bourns® products in such unauthorized applications is at the user's sole risk. Life-critical applications include devices identified by the U.S. Food and Drug Administration as Class III devices and generally equivalent classifications outside of the United States.

Bourns® standard products that are designed and tested for use in automotive applications will be described on the applicable data sheets as compliant with the applicable AEC-Q standard. Unless expressly and specifically approved in writing by two authorized Bourns representatives on a case-by-case basis, use of any other Bourns® standard products in an automotive application is not recommended, authorized or intended and will be at the user's sole risk.

Bourns® standard products are not tested to comply with United States Federal Aviation Administration standards generally or any other generally equivalent governmental organization standard applicable to products designed or manufactured for use in aircraft or space applications. Bourns® standard products that are designed and tested for use in aircraft or space applications will be described on the applicable data sheets as compliant with the RTCA DO-160 standard. Unless expressly and specifically approved in writing by two authorized Bourns representatives on a case-by-case basis, use of any other Bourns® standard product in an aircraft or space application is not recommended, authorized or intended and will be at the user's sole risk.

The use and level of testing applicable to Bourns® custom products shall be negotiated on a case-by-case basis by Bourns and the user for which such Bourns® custom products are specially designed. Absent a written agreement between Bourns and the user regarding the use and level of such testing, the provisions above applicable to Bourns® standard products shall also apply to such Bourns® custom products.

Users shall not sell, transfer, export or re-export any Bourns® products or technology for use in activities which involve the design, development, production, use or stockpiling of nuclear, chemical or biological weapons or missiles, nor shall they use Bourns® products or technology in any facility which engages in activities relating to such devices. The foregoing restrictions apply to all uses and applications that violate national or international prohibitions, including embargos or international regulations. Further, Bourns® products, technology or technical data may not under any circumstance be exported or re-exported to countries subject to international sanctions or embargoes, and Bourns® products may not, without prior authorization from Bourns and/or the U.S. Government, be resold, transferred, or re-exported to any party not eligible to receive U.S. commodities, software, and technical data.

To the maximum extent permitted by applicable law, Bourns disclaims (i) any and all liability arising out of the application or use of any Bourns® standard product, (ii) any and all liability, including, without limitation, special, punitive, consequential or incidental damages, and (iii) any and all implied warranties, including implied warranties of fitness for particular purpose, non-infringement and merchantability.

For your convenience, copies of this Legal Disclaimer Notice with German, Spanish, Japanese, Traditional Chinese and Simplified Chinese bilingual versions are available at:

Web Page: <http://www.bourns.com/legal/disclaimers-terms-and-policies>

PDF: <http://www.bourns.com/docs/Legal/disclaimer.pdf>

C1753 10/18/17

## RIWAYAT HIDUP PENULIS



Yudha Bintara Khresna Teguh Ary Arbajaya lahir di kota Surabaya pada tanggal 22 November 1997. Pada tahun 2018 penulis berusia 21 tahun. Yudha adalah panggilan akrabnya. Penulis merupakan anak kedua dari pasangan Alm. Teguh Tahan Udji dan Sudiyati. Penulis memeluk agama islam sejak lahir hingga saat ini. Penulis tinggal di jalan Kupang Gunung Timur 5A No. 1A Surabaya. Penulis sekolah di SDN Pakis VIII/375

Surabaya pada tingkat dasar, kemudian melanjutkan ke tingkat menengah di SMPN 4 Surabaya, dilanjutkan ke tingkat atas di SMAN 1 Surabaya, dan melanjutkan ke Institut Teknologi Sepuluh Nopember pada Departemen Teknik Elektro Otomasi. Penulis memiliki pengalaman organisasi sebagai panitia dalam sebuah acara nasional yang diadakan oleh Departemen Teknik Elektro Otomasi yaitu *Industrial Automation and Robotic Competition (IARC)* selama 2 Tahun.

E-mail : [yudhabintaraktaa@gmail.com](mailto:yudhabintaraktaa@gmail.com)

-----Halaman ini sengaja dikosongkan-----