



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 145561

**TELEMETRI *LEVEL* UNTUK TANGKI PENAMPUNGAN CAT
DAN PROSES PENGISIAN CAT PADA WADAH BERBASIS IOT
MENGUNAKAN ARDUINO**

Rudy Setiawan
NRP 2214030040
Miftahul Falahi Alfafa
NRP 2214030077

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng.

PROGRAM STUDI KOMPUTER KONTROL
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

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



FINAL PROJECT – TE 145561

***LEVEL TELEMETRY FOR PAINT STORAGE TANK AND
PAINT FILLING PROCESS ON CONTAINER BASED ON IOT
USING ARDUINO***

Rudy Setiawan
NRP 2214030040
Miftahul Falahi Alfafa
NRP 2214030077

Supervisor
Ir. Josaphat Pramudijanto, M.Eng.

***COMPUTER CONTROL STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2017***

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

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan tugas akhir saya dengan judul **“Telemetri Level Untuk Tangki Penampungan Cat Dan Proses Pengisian Cat Pada Wadah Berbasis IoT Menggunakan Arduino”** 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.

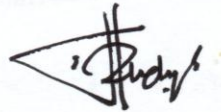
Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

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

Surabaya, 19 Juli 2017



Miftahul Falahi Alfafa
NRP 2214030077



Rudy Setiawan
NRP 2214030040

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



Institut
Teknologi
Sepuluh

TELEMETRI LEVEL UNTUK TANGKI PENAMPUNGAN CAT DAN PROSES PENGISIAN CAT PADA WADAH BERBASIS IOT MENGGUNAKAN ARDUINO



ITS

Institut
Teknologi
Sepuluh Nopember



ITS

Institut
Teknologi
Sepuluh N

ITS

Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR

ITS

Institut
Teknologi
Sepuluh Nopember

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada

Bidang Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Institut Teknologi Sepuluh Nopember

ITS

Institut
Teknologi
Sepuluh Nopember

Teknologi
Sepuluh Nopember

Menyetujui:

Dosen Pembimbing



Ir. Josaphat Pramudijanto, M.Eng.
NIP. 1962 10 05 1990 03 1003

ITS

Institut
Teknologi
Sepuluh Nopember

ITS

Institut
Teknologi
Sepuluh Nopember

ITS

Institut
Teknologi
Sepuluh Nopember

SURABAYA
JULI, 2017

ITS

Institut
Teknologi
Sepuluh Nopember

ITS

Institut

ITS

Institut
Teknologi

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

TELEMETRI *LEVEL* UNTUK PROSES PENGISIAN CAIRAN PADA BOTOL BERBASIS IOT MENGGUNAKAN ARDUINO

Nama : Rudy Setiawan
NRP : 2214 030 040
Nama : Miftahul Falahi Alfafa
NRP : 2214 030 077
Dosen Pembimbing : Ir. Josaphat Pramudijanto, MEng.
Nomor ID : 1962 10 05 1990 03 1003

ABSTRAK

SCADA (*Supervisory Control and Data Acquisition*) adalah sebuah sistem yang dirancang untuk sebuah pengendalian dan pengambilan data dalam pengawasan (Operator/Manusia). Telemetri merupakan suatu metode pengukuran yang dilakukan dari jarak yang relatif jauh. Telemetri sebenarnya adalah salah satu bentuk pengembangan teknologi telekomunikasi. Teknologi IoT (*Internet of Things*). Di era teknologi maju saat ini tidak dipungkiri bahwa proses otomasi yang dapat dimonitor secara jarak jauh sangatlah penting.

Cara kerja dari alat ini adalah tangki yang digunakan untuk menampung cairan cat yang sudah diberi *sensor level*. *Level* dari cairan dalam tangki dimonitor dan dijaga kondisinya agar tidak berlebih maupun kekurangan. Cairan cat dalam tangki dialirkan melalui pipa dan dikontrol untuk buka tutupnya menggunakan *solenoid valve*, terdapat wadah yang didistribusikan menggunakan konveyor dengan syarat botol berhenti tepat dibawah pipa aliran. Seluruh proses ini, dimonitoring menggunakan *Software LabVIEW* pada komputer di *Local Control Room*. Dan juga proses dari alat ini dimonitoring melalui ponsel dengan *Operating System Android* melalui WIFI dengan konsep IoT (*Internet of Things*).

Hasil yang didapatkan yaitu alat pengukur *level* cat pada tangki menggunakan *sensor level* dengan kondisi 0% adalah 0,56 *Volt* dan 100% adalah 4,22 *Volt*. Untuk mengotomatiskan proses pengisian cat kewadah ditambahkan konveyor dan *sensor* pendeteksi wadah, dimana ketika *sensor* tidak mendeteksi wadah 4,7 – 5 *Volt* dan ketika mendeteksi antara 2,6 – 3,9 *Volt*. Untuk monitoring menambahkan fitur HMI PC dan Android, namun mengalami *delay* dalam proses monitoring antara kondisi lapangan dan tampilan HMI selama 3,5 detik.

Kata Kunci : IoT, *Sensor Level*, *solenoid valve*, HMI, Android.

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

**LEVEL TELEMETRY FOR PAINT STORAGE TANK AND PAINT
FILLING PROCESS ON CONTAINER BASED ON IOT USING
ARDUINO**

Name : Rudy Setiawan
Register Number : 2214 030 040
Name : Miftahul Falahi Alfafa
Register Number : 2214 030 077
Supervisor : Ir. Josaphat Pramudijanto, MEng.
ID Number : 1962 10 05 1990 03 1003

ABSTRACT

SCADA (Supervisory Control and Data Acquisition) is a system designed for control and retrieval of data in surveillance (Operator / Human). Telemetry is a measurement method that is done from a relatively long distance. Telemetry is actually one form of telecommunication technology development. IoT Technology (Internet of Things). In the current era of advanced technology no doubt the process of automation that can be monitored remotely is very important.

The workings of this tool is a tank that is used to accommodate liquid that has been given sensors ten turn potentiometer. The level of fluid in the tank is monitored and maintained in order to avoid excess or deficiency. The liquid in the tank is passed through the pipe and controlled to open the lid using a solenoid valve. There is a container that is distributed using a conveyor provided that the bottle stops just below the flow pipe. The whole process is monitored using LabVIEW Software on a computer in the Local Control Room. And also the process of this tool is monitored via mobile phone with Android Operating System via WIFI with IoT concept (Internet of Things).

The result obtained is the level of paint on the tank level sensor using 0% is 0.56 Volt and 100% is 4.22 Volt. To automate the process of filling the paint, add a conveyor and container detection sensor, which is between 4.7 - 5 Volt when not detect a container and between 2.6 and 3.9 when detect. For monitoring the addition of HMI PC and Android features, but experienced a delay in the process of monitoring between the condition and display HMI for 3.5 seconds.

Keywords : IoT, Level Sensor, Solenoid Valve, HMI, Android.

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

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh.

Alhamdulillah segala puji syukur kami panjatkan kehadiran Allah SWT, karena atas limpahan rahmat dan hidayah-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul :

Telemetri *Level* Untuk Tangki Penampungan Cat Dan Proses Pengisian Cat Pada Wadah Berbasis IoT Menggunakan Arduino

Tugas Akhir ini disusun sebagai persyaratan untuk memperoleh gelar ahli madya teknik elektro pada bidang studi komputer kontrol departemen teknik elektro otomasi Institut Teknologi Sepuluh Nopember Surabaya.

Dalam penyusunan Tugas Akhir ini penulis banyak mendapat saran, dorongan, bimbingan serta keterangan-keterangan dari berbagai pihak yang merupakan pengalaman yang tidak dapat diukur secara materi, namun dapat membukakan mata penulis bahwa sesungguhnya pengalaman dan pengetahuan tersebut adalah guru yang terbaik bagi penulis. Oleh karena itu dengan segala hormat dan kerendahan hati perkenankanlah penulis mengucapkan terimakasih kepada :

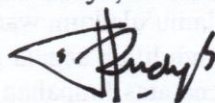
1. Kedua orang tua, yang selalu memberikan dukungan, semangat, dan doa kepada penulis.
2. Wanita istimewa yang pernah hadir dalam kehidupan penulis yang selalu menemani penulis saat suka dan duka dan tiada henti untuk tetap menyemangati.
3. Bapak Ir. Josaphat Pramudijanto, M.Eng. selaku Dosen pembimbing atas bantuan dan bimbingan hingga mata kuliah Tugas Akhir ini terselesaikan.
4. Keluarga besar *Andromeda* 2014 yang selalu memberikan bantuan yang tidak terkira terhadap penulis.

Dalam penyusunan tugas akhir ini, penulis menyadari masih terdapat banyak kekurangan yang dibuat baik sengaja maupun tidak sengaja, dikarenakan keterbatasan ilmu pengetahuan dan wawasan serta pengalaman yang penulis miliki. Untuk itu penulis *mohon* maaf atas segala kekurangan tersebut tidak menutup diri terhadap segala saran dan kritik serta masukan yang bersifat *konstruktif* bagi diri penulis.

Akhir kata semoga dapat bermanfaat bagi penulis sendiri, institusi pendidikan dan masyarakat luas. Amin!

Wassalamu 'alaikum Wr. Wb

Surabaya, 19 Juli 2017



Penulis

DAFTAR ISI

HALAMAN

HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	iii
PERNYA TAA N KEASLIAN TUGAS AKHIR	v
LEMBAR PENGESA HAN	vii
ABSTRAK	ix
<i>ABSTRACT</i>	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.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Sistematika Laporan.....	4
1.6 Relevansi.....	5
BAB II TEORI DASAR	7
2.1 SCADA	7
2.2 <i>Software Lab VIEW</i>	9
2.2.1 <i>Front Panel</i>	9
2.2.2 <i>Blog Diagram Dari Vi</i>	10
2.2.3 <i>Control dan Functions Pallete</i>	10
2.3 <i>Eihernet Shield Arduino Module</i>	11
2.4 <i>Photodiode</i>	13
2.5 Arduino Uno.....	14
2.5.1 Ringkasan.....	15
2.5.2 Daya	15
2.5.3 Memori	16
2.5.4 <i>Input dan Output</i>	16
2.5.5 Komunikasi	17
2.6 <i>Web hosting</i>	17
2.7 <i>Software Android Studio</i>	18
2.8 <i>phpMyAdmin</i>	19
2.9 <i>Wemos D1 Mini</i>	20

BAB III PERANCANGAN SISTEM	23
3.1 Blok Diagram Cara Kerja Alat	23
3.1.1 Cara Kerja Dari Sistem	23
3.1.2 Cara Kerja Dari Sistem Proses Pengisian Cat	24
3.2 Perencanaan dan Pembuatan <i>Hardware</i>	24
3.3 Perancangan Sistem Arduino	27
3.4 Perancangan Sistem <i>Sensor</i> Pendeteksi Wadah	30
3.5 Perancangan Sistem <i>Sensor Level</i>	31
3.5.1 Rangkaian <i>Wheatstone Bridge</i> Dan ASC.....	32
3.5.2 Perhitungan Rangkaian <i>Wheatstone Bridge</i> dan ASC.....	34
3.6 Perancangan Sistem Kendali Motor DC	37
3.7 Perancangan Sistem <i>Solenoid Valve</i>	37
3.8 Perancangan Sistem Pompa	38
3.9 Perencanaan dan Pembuatan <i>Software</i>	39
3.9.1 HMI (<i>Human Machine Interface</i>) Untuk PC.....	39
3.9.2 HMI (<i>Human Machine Interface</i>) Untuk Android	39
3.9.3 Perancangan <i>Software</i> Arduino	41
3.9.4 Perancangan <i>Software Sensor Photodiode</i>	44
3.9.5 Perancangan <i>Software Sensor Level</i>	45
3.9.6 Perancangan <i>Software</i> Motor DC.....	45
3.9.7 Perancangan <i>Software Solenoid Valve</i> dan Pompa.....	49
3.9.8 Perancangan <i>Software Shield Ethernet</i> Arduino.....	49
3.9.9 Perancangan <i>Software Wemos D1 Mini</i>	49
3.9.10 Perancangan <i>Website</i>	49
3.9.11 Perancangan Sistem <i>Web hosting</i>	54
3.9.12 Perancangan Sistem <i>phpMyAdmin</i>	55
BAB IV HASIL SIMULASI DAN UJI COBA	59
4.1 Pengujian <i>Sensor level</i>	59
4.1.1 Data Hasil Pengukuran	59
4.2 Pengujian <i>Sensor Photodiode</i>	60
4.3 Pengujian <i>Solenoid Valve</i>	63
4.4 Pengujian Motor DC	65
4.5 Pengujian Pompa	67
4.6 Pengujian HMI Android	68
4.7 Pengujian HMI Lab VIEW	69
4.8 Pengujian <i>Shield Ethernet</i>	71
4.9 Pengujian <i>Wemos D1 Mini</i>	72
4.10 Pengujian <i>Web hosting</i>	73

BAB V PENUTUP	75
5.1 Kesimpulan	75
5.2 Saran	76
DAFTAR PUSTAKA	77
LAMPIRAN A	79
A.1 Lampiran Program Arduino	79
A.2 Lampiran Program <i>Wemos D1 Mini</i>	87
A.3 Lampiran Program Android	91
A.4 Lampiran Program <i>Update Data Web hosting</i>	103
A.5 Lampiran Program <i>Login Android</i>	105
A.6 Lampiran Program Koneksi ke <i>Database di Web hosting</i>	106
A.7 Lampiran Blok Diagram <i>Sensor Level</i>	107
A.8 Lampiran Blok Diagram <i>Sensor</i> Pendeteksi Wadah	108
A.9 Lampiran Blok Fungsional <i>Sensor Level</i>	109
A.10 Lampiran Blok Diagram <i>Sensor</i> Pendeteksi Wadah	110
LAMPIRAN B	111
B.1 Dokumentasi Keseluruhan Alat Dari Proses Pengisian Cairan .	111
B.2 Dokumentasi Tampilan HMI.....	111
B.3 Dokumentasi Tangki Penampungan.....	112
B.4 Dokumentasi Meja Untuk Kontroler Alat Pengisian Cairan	112
B.5 Dokumentasi Kontroller Untuk Alat Pengisian Cairan.....	113
B.6 Dokumentasi Konveyor Untuk Distribusi Wadah	113
LAMPIRAN C.....	115
C.1 <i>Datasheet Arduino Uno</i>	115
C.2 <i>Datasheet Wemos D1 Mini</i>	116
C.3 <i>Datasheet Ethernet</i>	117
LAMPIRAN D	119

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

DAFTAR GAMBAR

HALAMAN

Gambar 2.1	<i>Primitive SCADA</i>	8
Gambar 2.2	<i>Modern SCADA</i>	8
Gambar 2.3	Tampilan <i>Front panel</i>	9
Gambar 2.4	Blok Diagram.....	10
Gambar 2.5	<i>Control Pallette</i>	10
Gambar 2.6	<i>Functions Pallette</i>	11
Gambar 2.7	Modul <i>Ethernet Arduino</i>	12
Gambar 2.8	Tanggapan Frekuensi <i>Sensor Photodiode</i>	13
Gambar 2.9	Hubungan Keluaran <i>Photodiode</i> Dengan Intensitas Cahaya.....	14
Gambar 2.10	Arduino UNO	15
Gambar 2.11	Tampilan Antarmuka Android Studio	19
Gambar 2.12	Tampilan <i>phpMyAdmin</i>	20
Gambar 2.13	<i>Wemos D1 Mini</i>	20
Gambar 3.1	Rancangan Mekanik Pada Proses Pengisian Cairan	25
Gambar 3.2	Rancangan Mekanik Pada Proses Pengisian Cairan	25
Gambar 3.3	Rancangan Mekanik Pada Konveyor.....	26
Gambar 3.4	Rancangan Mekanik Pada Konveyor.....	26
Gambar 3.5	Rancangan Mekanik Pada Tangki Penampungan.....	27
Gambar 3.6	Rancangan Mekanik Pada Tangki Penampungan.....	27
Gambar 3.7	Diagram Blok Sistem Pengisian Cairan Secara Otomatis.....	29
Gambar 3.8	Perancangan Pemasangan <i>Photodiode Sensor</i>	30
Gambar 3.9	Rangkaian Elektronika <i>Photodiode Sensor</i> Bagian Pertama	31
Gambar 3.10	Rangkaian Elektronika <i>Photodiode Sensor</i> Bagian Kedua	31
Gambar 3.11	Perancangan <i>Sensor Ten Turn Potentiometer</i> di Tangki Penampungan.....	32
Gambar 3.12	Rangkaian <i>Wheatstone</i> Untuk Mengetahui Beda Potensial.....	33
Gambar 3.13	Rangkaian <i>Wheatstone</i> Dengan Pin TO ASC	33
Gambar 3.14	Rangkaian <i>Analog Signal Conditioning</i>	36
Gambar 3.15	Rangkaian <i>Driver Motor DC</i>	37

Gambar 3.16	Rangkaian Elektronika <i>Solenoid Valve</i>	38
Gambar 3.17	Rangkaian Elektronika Pompa AC 220 Volt	38
Gambar 3.18	HMI (<i>Human Machine Interface</i>) Untuk PC (<i>Personal Computer</i>).....	39
Gambar 3.19	Blok Fungsional Sistem.....	40
Gambar 3.20	HMI (<i>Human Machine Interface</i>) Untuk Android <i>Phone</i>	41
Gambar 3.21	Tampilan <i>Software</i> Arduino	42
Gambar 3.22	Tampilan Arduino Ketika Tidak Terjadi <i>Error</i> Pada Saat <i>Verify</i> Program	43
Gambar 3.23	Memilih <i>Board</i> Arduino Pada <i>Software</i> Arduino IDE....	44
Gambar 3.24	Memilih <i>Port</i> Pada Arduino	44
Gambar 3.25	<i>Flowchart</i> <i>Sensor Photodiode</i>	45
Gambar 3.26	<i>Flowchart</i> Untuk <i>Sensor Level</i>	46
Gambar 3.27	<i>Flowchart</i> Motor Konveyor.....	47
Gambar 3.28	<i>Flowchart</i> Motor Pendorong	48
Gambar 3.29	<i>Flowchart</i> <i>Solenoid Valve</i> dan Pompa Tangki Utama	50
Gambar 3.30	<i>Flowchart</i> <i>Solenoid Valve</i> dan Pompa Wadah	51
Gambar 3.31	<i>Flowchart</i> <i>Shield Ethernet</i>	52
Gambar 3.32	<i>Flowchart</i> <i>Wemos D1 Mini</i>	53
Gambar 3.33	Tampilan <i>Subdomain</i>	54
Gambar 3.34	Tampilan Data <i>Web hosting</i>	54
Gambar 3.35	Kontrol Panel <i>Website</i>	55
Gambar 3.36	<i>Flowchart</i> <i>Web hosting</i>	56
Gambar 3.37	Tampilan <i>Database phpMyAdmin</i>	57
Gambar 4.1	<i>Sensor Photodiode</i> Dengan Ada Wadah di Konveyor....	61
Gambar 4.2	<i>Sensor Photodiode</i> Dengan Tidak Ada Wadah di Konveyor.....	61
Gambar 4.3	Rangkaian Untuk Pengujian <i>Sensor</i> Pendeteksi Wadah..	62
Gambar 4.4	<i>Solenoid Valve</i>	64
Gambar 4.5	Rangkaian Untuk Pengujian <i>Solenoid Valve</i>	64
Gambar 4.6	Motor DC Sebagai Penggerak Konveyor.....	65
Gambar 4.7	Motor DC Sebagai Pendorong Wadah.....	66
Gambar 4.8	Rangkaian Untuk Pengujian Motor DC	66
Gambar 4.9	Rangkaian Pengujian Untuk Pompa	67
Gambar 4.10	Tampilan HMI Android.....	68
Gambar 4.11	Diagram Blok HMI Lab VIEW	70

Gambar 4.12	Tampilan <i>OPC Client</i> Untuk Mengetahui Status Pengiriman Data Pada Modul <i>Shield Ethernet</i>	71
Gambar 4.13	Kode Untuk Mengirimkan Nilai <i>Sensor Level</i> ke HMI LabVIEW	71
Gambar 4.14	Kode Untuk Mengirimkan Data dari Modul <i>Wemos</i> ke <i>Web hosting</i>	72
Gambar 4.15	Tabel <i>Database</i> Pada <i>Web hosting</i>	73

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

DAFTAR TABEL

HALAMAN

Tabel 2.1	Spesifikasi <i>Ethernet Arduino</i>	11
Tabel 3.1	<i>Data Mapping</i> I/O Arduino UNO	28
Tabel 3.2	Data <i>Wheatstone Bridge</i> Yang Diinginkan	32
Tabel 3.3	Data <i>Analog Signal Conditioning</i> Yang Diinginkan	33
Tabel 4.1	Data Hasil Pengukuran Pada <i>Output Wheatstone Bridge</i> ...	59
Tabel 4.2	Data Hasil Pengukuran Pada <i>Output Analog Signal Conditioning</i>	59
Tabel 4.3	Data <i>Sensor</i> Potensiometer Untuk Pengukuran <i>Level Air</i> ..	60
Tabel 4.4	Data <i>Sensor Photodiode</i> Untuk Botol Akrilik Tanpa Air ...	62
Tabel 4.5	Data <i>Sensor Photodiode</i> Untuk Botol Akrilik Berisi Air....	63
Tabel 4.6	Data <i>Sensor Photodiode</i> Untuk Botol Akrilik Yang Dilapisi Stiker Warna <i>Orange</i>	63
Tabel 4.7	Pengujian <i>Solenoid Valve</i>	65
Tabel 4.8	Pengujian Motor DC.....	67
Tabel 4.9	Hasil Pengujian Pompa	68
Tabel 4.10	Pengujian Konektivitas HMI Android	69
Tabel 4.11	Pengujian Konektivitas HMI Lab VIEW	69
Tabel 4.12	Pengujian <i>Shield Ethernet</i>	71
Tabel 4.13	Pengujian Modul <i>Wemos D1 Mini</i>	72
Tabel 4.14	Pengujian <i>Web hosting</i>	74

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

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada zaman *modern* ini penggunaan cat sangatlah dibutuhkan. Misalnya untuk memperindah rumah, mobil, dan lain lain, sehingga saat ini banyak bermunculan industri cat untuk memenuhi kebutuhan masyarakat. Namun disini timbul permasalahan, yaitu habisnya persediaan cat yang akan dikemas karena tangki yang digunakan untuk menampung cat dipantau *level*nya oleh manusia. Selain itu, untuk melihat *level* tangki, pekerja harus berada di lokasi tangki penampungan. Dari permasalahan tersebut, industri cat mempunyai solusi dengan membuat tangki penampung cat secara otomatis, cara kerjanya adalah ketika persediaan cat dalam tangki habis maka pompa akan mengisi cat didalamnya dan ketika penuh pompa akan berhenti mengisi. Otomatisasi sistem pendeteksi cat tersebut memerlukan komponen *sensor*. *Sensor* ini yang akan memberikan parameter untuk mengatur nyala dan mati pompa.

Untuk dapat membuat *sensor* bisa mengatur pompa tersebut diperlukan pembacaan nilai *sensor*. Pembacaan *sensor* sendiri dilakukan dengan menghubungkan *sensor* dengan kontroler (modul kontroler bawaan *sensor*), kemudian modul dihubungkan dengan komputer atau laptop. Namun terdapat permasalahan, dimana sistem yang disebut otomatis masih memerlukan manusia untuk datang ke lokasi tangki untuk memantau nilai *sensor*. Oleh sebab itu pada sistem yang dibuat otomatis, proses pengukuran nilai *sensor* sebaiknya menggunakan metode telemetri. Telemetri adalah suatu metode pengukuran yang dilakukan dari jarak yang relatif jauh. Telemetri sebenarnya merupakan salah satu pengembangan teknologi telekomunikasi. Proses pembacaan nilai *sensor* dilakukan di ruang kontrol yang letaknya jauh dari lokasi tangki penampungan.

Proses pengisian cat diawali dengan mengisi cat kedalam wadah, lalu wadah yang sudah terisi akan menuju ke proses tahap selanjutnya. Proses pengisian cat dilakukan secara otomatis dan hanya dapat dimonitor melalui ruang kontrol lokal yang letaknya berdekatan dengan lokasi tersebut. Karena sistem pengisian cat dan monitoringnya dilakukan secara otomatis, maka jumlah pekerja menjadi sedikit sehingga lebih efisien. Fungsi dari ruang kontrol lokal adalah untuk

memberitahu pekerja dibagian proses pengisian jika ada kegagalan sistem pada proses tersebut. Jika terdapat kegagalan sistem maka alarm akan berbunyi sehingga petugas di ruang kontrol dapat segera melakukan tindakan. Bagaimana jika orang yang berada di ruang kontrol sedang melakukan perbaikan atau pengecekan di lapangan, sedangkan sistem mengalami kegagalan dan hanya ruang kontrol lokal yang mendapat pemberitahuan tersebut. Sistem pada bagian proses pengisian cat akan otomatis berhenti akibat kegagalan tersebut. Lalu membuat bagian setelah proses pengisian cat terhenti juga dan pekerja di bagian tersebut tidak tahu mengapa sistem prosesnya berhenti, tetapi pada sistem bagian tersebut tidak memberitahukan adanya kegagalan. Hal yang dilakukan pada bagian lain hanya menunggu *konfirmasi* bahwa ada kegagalan di proses pengisian cat atau mencari pada bagian mana yang terjadi kegagalan. Untuk itu diperlukan suatu metode pemantauan jarak jauh untuk proses pengisian cat menggunakan metode telemetri. Sehingga proses pengisian cat tidak hanya dipantau oleh petugas lapangan lokal saja, tetapi bisa juga dipantau oleh anggota dari bagian lain tanpa harus berada di ruang kontrol lokal proses pengisian cat. Sehingga jika terjadi kegagalan pada sistem proses pengisian cat, maka anggota dari bagian lain dapat mengetahui atau memberitahukan secara langsung kepada petugas lapangan melalui *telepon* untuk segera melakukan pengecekan atau perbaikan pada bagian yang bermasalah.

Dengan menggunakan sistem telemetri maka seluruh bagian dapat melihat informasi dari bagian yang lain pada industri melalui HMI (*Human Machine Interface*) lokal. Sehingga jika terjadi kegagalan pada salah satu bagian dan pekerja yang bertugas tidak tahu akan kegagalan tersebut, maka petugas pada bagian lain dapat memberitahukan kegagalan yang terjadi. Tetapi bagaimana jika pimpinan ingin memonitor jalannya proses produksi dan sedang jauh dari lokasi ruang kontrol lokal atau industri. Ditambah lagi jika pimpinan ingin mengetahui jumlah produksi pada proses pengisian cat, maka pimpinan harus bertanya terlebih dahulu kepada petugas lapangan. Hal tersebut cukup merepotkan dan tidak efisien, karena membutuhkan waktu cukup lama untuk perjalanan dan ditambah lagi jika petugas lapangan salah menginformasikan tentang proses produksi. Untuk itu diperlukan HMI yang dapat diakses dari mana saja dan mudah untuk digunakan. Oleh sebab itu dibuatlah HMI untuk ponsel berbasis Android yang terhubung *internet*, yang mana jaringan *internet* tersebut menggunakan teknologi IoT (*Internet of Things*). IoT sendiri adalah suatu teknologi yang menghubungkan data-data dari *hardware* atau mesin ke *internet*.

Dengan adanya teknologi IoT maka seluruh informasi atau data-data tentang proses produksi langsung secara otomatis dikirim ke *web*, lalu data-data tersebut dikirim ke aplikasi HMI pada ponsel Android.

Arduino adalah salah satu komponen IoT yang dapat diaplikasikan sebagai pengendali jarak jauh dengan jaringan *internet* dengan bantuan modul WIFI untuk Arduino. Modul WIFI merupakan modul yang memungkinkan Arduino terkoneksi dengan *internet* sehingga dapat diakses melalui *website* dengan protokol TCP/IP (*Transmission Control Protocol / Internet Protocol*).

1.2 Permasalahan

- a. Dikarenakan sistem kerja dari tangki penampungan secara otomatis dan lokasinya yang jauh dari ruang kontrol lokal, maka diperlukan monitoring *level* tangki dari jarak jauh. Penanggung jawab untuk *Point* ini adalah Miftahul Falahi Alfafa.
- b. Dikarenakan proses pengisian cat dilakukan secara otomatis dan hanya dapat dimonitor melalui ruang kontrol lokal yang letaknya berdekatan dengan lokasi tersebut, maka diperlukan monitoring jalannya proses pengisian cat dari jarak jauh. Penanggung jawab untuk *Point* ini adalah Rudy Setiawan.
- c. Dikarenakan proses pemantauan hanya melalui ruang kontrol lokal, maka diperlukan pemantauan proses produksi dari jarak jauh (jauh dari ruang kontrol lokal) Penanggung jawab pada *Point* ini adalah Miftahul Falahi Alfafa dan Rudy Setiawan.

1.3 Batasan Masalah

Dari perumusan masalah di atas, ada beberapa hal yang perlu dibatasi sehingga penelitian yang dilakukan dapat tercapai. Batasan masalah dalam pengerjaan Tugas Akhir ini, yaitu:

- a. Dalam penelitian ini untuk mengukur *volume* cat dalam tangki penampungan menggunakan *sensor Ten Turn Potentiometer*.
- b. Dalam penelitian ini hanya melakukan proses pemantuan ketinggian cat di tangki penampungan, dan melakukan proses pengisian cat kedalam wadah saja, tidak sampai ke proses pengemasannya.
- c. Untuk pengujian proses pengisian wadah, dilakukan di dalam ruangan. (tidak terkena sinar matahari secara langsung).
- d. Wadah yang digunakan sebagai tempat cat pada proses pengisian berbentuk balok ukuran $(p * l * t) = (8 * 8 * 15)$.

- e. Untuk motor penggerak dan pendorong pada konveyor menggunakan kontrol *ON OFF* saja
- f. Kontroler yang digunakan adalah Arduino UNO
- g. HMI yang digunakan ada 2 macam, yaitu untuk PC/Laptop menggunakan *software* LabVIEW, dan untuk ponsel Android menggunakan Android Studio

Dengan adanya batasan masalah ini diharapkan hasil akhir dari Tugas Akhir ini dapat tercapai.

1.4 Tujuan

- a. Merancang dan membuat alat untuk mengukur ketinggian cat dalam tangki penampungan cat dengan menggunakan *sensor level* yang dipadukan sistem SCADA telemetri. (Penanggung jawab Miftahul Falahi Alfafa)
- b. Merancang dan membuat alat untuk proses pengisian cat pada wadah secara otomatis yang dipadukan sistem SCADA telemetri. (Penanggung jawab Rudy Setiawan)
- c. Merancang dan membuat sistem pemantauan (HMI) untuk PC (*Personal Computer*) menggunakan *software* LabVIEW dan Android berbasis IoT menggunakan *software* Android Studio. Miftahul Falahi Alfafa bertanggung jawab untuk HMI Android berbasis IoT, dan Rudy Setiawan bertanggung jawab untuk HMI PC dengan *software* LabVIEW

1.5 Sistematika Laporan

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

Bab I Pendahuluan

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

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka yang menunjang dalam pengerjaan alat ini. Bab ini dibagi menjadi dua bagian dimana Rudy Setiawan mencari bahan dari materi pada Poin 2.1 sampai dengan Poin 2.7. dan Miftahul Falahi Alfafa mencari bahan dari materi pada Poin 2.8 sampai dengan Poin 2.14.

Bab III Perancangan Sistem

Pada bab ini akan dibahas mengenai perancangan sistem dari alat yang akan dibuat. Rudy Setiawan bertugas untuk merancang dan membuat *hardware* yang berhubungan dengan konveyor, lalu merancang dan membuat *software* HMI untuk PC menggunakan *software* LabIEW, dan juga merangkai rangkaian elektronika yang mendukung dengan jalannya konveyor. Miftahul Falahi Alfafa bertugas untuk merancang dan membuat *hardware* yang berhubungan dengan tangki penampungan cairan, lalu merancang dan membuat *software* HMI untuk Android menggunakan *software* Android Studio, dan juga merangkai rangkaian elektronika yang mendukung dengan jalannya sistem pengukuran *level* tangki penampungan.

Bab IV Simulasi, Implementasi dan Uji Coba

Bab ini akan dijelaskan mengenai uji coba pada masing-masing komponen dan juga data dari komponen. Pada bab ini Rudy Setiawan bertanggung jawab pada pegujian yang berhubungan dengan konveyor dari *sensor* maupun HMI. Begitu juga dengan Miftahul Falahi Alfafa bertanggung jawab pada pengujian yang berhubungan dengan Tangki penampungan dari *sensor* maupun HMI. Tiap Poin dijelaskan siapa yang bertanggung jawab

Bab V Penutup

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

1.6 Relevansi

Adanya telemetri *level* diharapkan dapat membantu proses monitoring pada *level* tangki penampungan cat yang sulit untuk dilakukan pengukuran *level* secara *manual* Dengan terbentuknya alat ini secara otomatis diharapkan dapat mempermudah pekerjaan manusia dalam proses pengisian cat sesuai yang diinginkan. Dengan adanya *software* yang dapat memonitor (HMI berbasis IoT) diharapkan dapat

membantu perusahaan atau industri dalam memantau kegiatan produksi pengisian cat kedalam wadah.

BAB II

TEORI DASAR

Pada bab ini akan dibahas mengenai tinjauan pustaka dimana ada 9 Poin. Tinjauan pustaka ini dibagi menjadi 2 bagian tiap masing masing mahasiswa, Rudy Setiawan mencari bahan dari materi pada Poin 2.1 sampai dengan Poin 2.5. dan Miftahul Falahi Alfafa mencari bahan dari materi pada Poin 2.6 sampai dengan Poin 2.9. dimana tiap materi yang dicari sebagai dasar materi untuk pengerjaan alat yang dibuat masing-masing mahasiswa, dan juga sebagai dasar materi untuk pembuatan keseluruhan alat ini.

2.1 SCADA [1]

SCADA atau *Supervisory Control and Data Acquisition* adalah sebuah sistem yang dirancang untuk sebuah pengendalian dan pengambilan data dalam pengawasan (Operator/Manusia). biasanya SCADA digunakan untuk pengendalian suatu proses pada industri.

Ada dua elemen dalam Aplikasi SCADA, yaitu:

1. Proses, sistem, mesin yang akan dipantau dan dikontrol - bisa berupa *power plant*, sistem pengairan, jaringan komputer, sistem lampu trafik lalu-lintas atau apa saja.
2. Sebuah jaringan peralatan ‘cerdas’ dengan antarmuka ke sistem melalui *sensor* dan luaran kontrol. Dengan jaringan ini, yang merupakan sistem SCADA, membolehkan Anda melakukan pemantauan dan pengontrolan komponen-komponen sistem tersebut.

Anda dapat membangun sistem SCADA menggunakan berbagai macam teknologi maupun protokol yang berbeda-beda.

SCADA (*Supervisory Control And Data Acquisition*) umumnya mengacu pada sistem kontrol industri: sistem komputer yang memantau dan mengontrol industri, infrastruktur, atau fasilitas berbasis proses. Terdapat perbedaan “*primitive*” SCADA (lihat pada Gambar 2.1) dan “*modern*” SCADA (lihat pada Gambar 2.2)



Gambar 2.1 *Primitive SCADA*



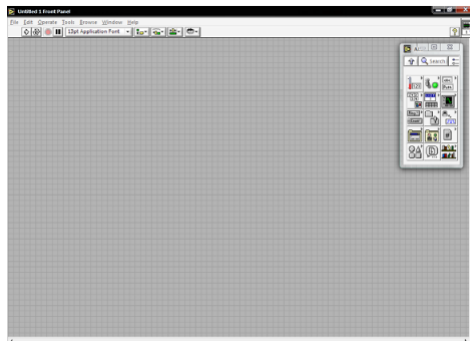
Gambar 2.2 *Modern SCADA*

2.2 Software LabVIEW [2]

LabVIEW adalah sebuah *software* pemrograman yang diproduksi oleh National Instruments dengan konsep yang berbeda. Seperti bahasa pemrograman lainnya yaitu C++, Matlab atau Visual basic, LabVIEW juga mempunyai fungsi dan peranan yang sama, perbedaannya bahwa LabVIEW menggunakan bahasa pemrograman berbasis grafis atau blok diagram sementara bahasa pemrograman lainnya menggunakan basis text. Program LabVIEW dikenal dengan sebutan Vi atau *Virtual instruments* karena penampilannya dan operasinya dapat meniru sebuah *instrument*. Pada LabVIEW, *user* pertama-tama membuat *user interface* atau *front panel* dengan menggunakan *control* dan indikator, yang dimaksud dengan kontrol adalah *knobs*, *push buttons*, *dials* dan peralatan *input* lainnya sedangkan yang dimaksud dengan indikator adalah *graphs*, LEDs dan peralatan display lainnya. Setelah menyusun *user interface*, lalu *user* menyusun blok diagram yang berisi kode-kode VIs untuk mengontrol *front panel*. *Software* LabVIEW terdiri dari tiga komponen utama, yaitu :

2.2.1 Front Panel

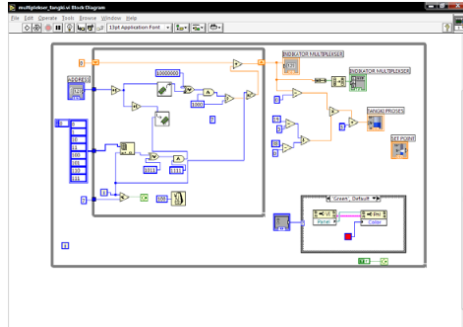
front panel adalah bagian *window* yang berlatar belakang abu-abu serta mengandung *control* dan indikator. *front panel* digunakan untuk membangun sebuah VI, menjalankan program dan *mendebug* program. Tampilan dari *front panel* dapat di lihat pada Gambar 2.3



Gambar 2.3 Tampilan *Front panel*

2.2.2 Blok Diagram Dari Vi

Blok diagram adalah bagian *window* yang berlatar belakang putih berisi *source code* yang dibuat dan berfungsi sebagai instruksi untuk *front panel*. Tampilan dari blok diagram dapat dilihat pada Gambar 2.4.



Gambar 2.4 Blok Diagram

2.2.3 Control dan Functions Palette

Control dan *Functions Palette* digunakan untuk membangun sebuah Vi.

1. Control Palette

Control Palette merupakan tempat beberapa *control* dan indikator pada *front panel*, *control palette* hanya tersedia di *front panel*. (lihat pada Gambar 2.5)



Gambar 2.5 Control Palette

2. *Functions Pallette*

Functions Pallette digunakan untuk membangun sebuah blok diagram, *functions pallette* hanya tersedia pada blok diagram. (lihat pada Gambar 2.6)



Gambar 2.6 *Functions Pallette*

2.3 *Ethernet Shield Arduino Module* [3]

Ethernet Shield menambahkan kemampuan arduino board agar terhubung ke jaringan komputer. *Ethernet shield* berbasiskan *chip ethernet Wiznet W5100*. *Ethernet library* digunakan dalam menulis program agar arduino board dapat terhubung ke jaringan dengan menggunakan arduino *ethernet shield*.

Pada *ethernet shield* terdapat sebuah *slot micro-SD*, yang dapat digunakan untuk menyimpan *file* yang dapat diakses melalui jaringan. *Onboard micro-SD card reader* diakses dengan menggunakan *SD library*. Arduino board berkomunikasi dengan W5100 dan *SD card* menggunakan *bus SPI (Serial Peripheral Interface)*. Komunikasi ini diatur oleh *library SPI.h* dan *Ethernet.h* Adapun spesifikasi yang dimiliki *Ethernet Arduino* diperlihatkan seperti pada Tabel 2.1.

Tabel 2.1 Spesifikasi *Ethernet Arduino*

<i>Microcontroller</i>	<i>Atmega328</i>
<i>Operating Voltage</i>	5V
<i>Input Voltage (recommended)</i>	7-12V
<i>Input Voltage (Limit)</i>	6-20V
<i>Digital I/O Pins</i>	14 (of which 4 provide PWM output)
<i>Arduino Pins Reserved</i>	
	10 tp 13 used for SPI
	4 used for SD Card
	2 w5199 interrupt (when bridged)

<i>Microcontroller</i>	<i>Atmega328</i>
<i>Analog Input Pins</i>	6
<i>DC Current per I/O Pin</i>	40 mA
<i>DC Current for 3.3V Pin</i>	50 mA
<i>Flash Memory</i>	32 KB of which 0,5 KB used by bootloader
<i>SRAM</i>	2 KB
<i>EEPROM</i>	1 KB
<i>Clock Speed</i>	16 MHz
<i>W5100 TCP/IP Embedded Ethernet Controller</i>	
<i>Power Over Ethernet ready Magnetic Jack</i>	
<i>Micro SD card, with active Voltage translator</i>	
<i>Length</i>	68,6 mm
<i>Width</i>	53,3 mm
<i>Weight</i>	28 g

Hardware ethernet biasanya berupa modul yang dapat dihubungkan ke *port* komputer (*parallel*, *serial*, dan *USB*). Salah satunya *product ethernet* yang dipakai untuk Tugas Akhir ini adalah modul *Ethernet* dari Arduino yaitu *Ethernet Shield* Arduino seperti pada Gambar 2.7.



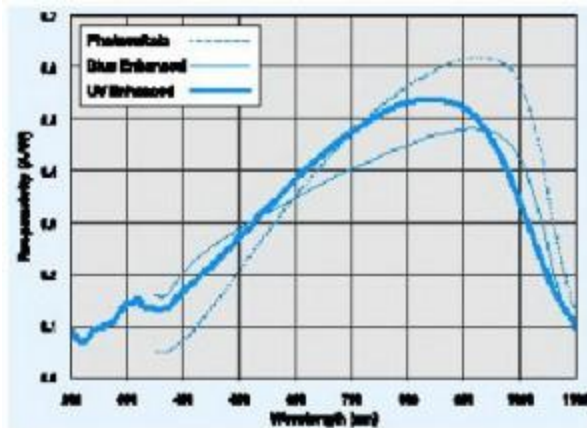
Gambar 2.7 Modul *Ethernet* Arduino

2.4 Photodiode [4]

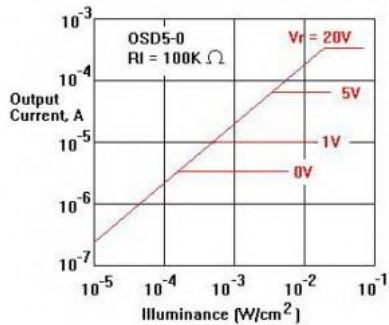
Sensor photodiode merupakan dioda yang peka terhadap cahaya, *sensor photodiode* akan mengalami perubahan resistansi pada saat menerima intensitas cahaya dan akan mengalirkan arus listrik secara *forward* sebagaimana dioda pada umumnya. *Sensor photodiode* adalah salah satu jenis *sensor* peka cahaya (*photodetector*). Jenis *sensor* peka cahaya lain yang sering digunakan adalah *phototransistor*. *Photodiode* akan mengalirkan arus yang membentuk fungsi *linear* terhadap intensitas cahaya yang diterima. Arus ini umumnya teratur terhadap *power density* (D_p). Perbandingan antara arus keluaran dengan *power density* disebut sebagai *current responsivity*. Arus yang dimaksud adalah arus bocor ketika *photodiode* tersebut disinari dan dalam keadaan dipanjar mundur.

Tanggapan frekuensi *sensor photodiode* tidak luas. Dari rentang tanggapan itu, *sensor photodiode* memiliki tanggapan paling baik terhadap cahaya infra merah, tepatnya pada cahaya dengan panjang gelombang sekitar $0,9 \mu\text{m}$. Kurva tanggapan *sensor photodiode* ditunjukkan pada Gambar 2.8.

Hubungan antara keluaran *sensor photodiode* dengan intensitas cahaya yang diterimanya ketika dipanjar mundur adalah membentuk suatu fungsi yang linier. Hubungan antara keluaran *sensor photodiode* dengan intensitas cahaya ditunjukkan pada Gambar 2.9.



Gambar 2.8 Tanggapan Frekuensi *Sensor Photodiode*



Gambar 2.9 Hubungan Keluaran *Photodiode* Dengan Intensitas Cahaya

2.5 Arduino Uno [5]

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

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

Nama "Uno" berarti satu dalam bahasa Italia, untuk menandai peluncuran Arduino 1.0. Uno dan versi 1.0 akan menjadi versi referensi dari Arduino. Uno adalah yang terbaru dalam serangkaian *board* USB Arduino, dan sebagai model referensi untuk *platform* Arduino, untuk perbandingan dengan versi sebelumnya, lihat indeks *board* Arduino pada Poin 2.5.1. untuk *board* arduino lihat pada Gambar 2.10



Gambar 2.10 Arduino UNO

2.5.1 Ringkasan

1. *Microcontroller* ATmega328
2. Operasi dengan daya *5Volt*
3. *Input* Tegangan (disarankan) 7-12V
4. *Input* Tegangan (batas) 6-20V
5. *Digital I/ O* Pin 14 (dimana 6 memberikan *output* PWM)
6. *Analog Input* Pin 6
7. DC Lancar per I/ O Pin 40 mA
8. Saat 3,3V Pin 50 mA DC
9. *Flash Memory* 32 KB (ATmega328) yang 0,5 KB digunakan oleh *bootloader*
10. SRAM 2 KB (ATmega328)
11. EEPROM 1 KB (ATmega328)
12. *Clock Speed* 16 MHz

2.5.2 Daya

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya *eksternal* (otomatis).

Eksternal (non-USB) daya dapat berasal baik dari AC-ke adaptor-DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan *plug jack* pusat-positif ukuran 2.1 mm konektor *POWER*. Ujung kepala dari baterai dapat dimasukkan ke dalam Gnd dan Vin pin *header* dari konektor *POWER*.

Kisaran kebutuhan daya yang disarankan untuk *board* Uno adalah 7 sampai dengan 12 *Volt*, jika diberi daya kurang dari 7 *Volt* kemungkinan pin 5V Uno dapat beroperasi tetapi tidak stabil kemudian jika diberi daya lebih dari 12V, regulator tegangan bisa panas dan dapat merusak *board* Uno.

Pin listrik adalah sebagai berikut:

1. VIN. Tegangan masukan kepada *board* Arduino ketika itu menggunakan sumber daya eksternal (sebagai pengganti dari 5 Volt koneksi USB atau sumber daya lainnya).
2. 5V. Catu daya digunakan untuk daya mikrokontroler dan komponen lainnya.
3. 3v3. Sebuah pasokan 3,3 Volt dihasilkan oleh *regulator on-board*.
4. GND. *Ground* pin.

2.5.3 Memori

ATmega328 memiliki 32 KB (dengan 0,5 KB digunakan untuk *bootloader*), 2 KB dari SRAM dan 1 KB EEPROM (yang dapat dibaca dan ditulis dengan EEPROM *library*).

2.5.4 Input dan Output

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

1. Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini dihubungkan ke pin yang berkaitan dengan *chip* Serial ATmega8U2 USB-to-TTL.
2. Eksternal menyela: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, dengan batasan tepi naik atau turun, atau perubahan nilai. Lihat (*attachInterrupt*) fungsi untuk rincian lebih lanjut.
3. PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan *output* PWM 8-bit dengan fungsi `analogWrite()`.
4. SPI: 10 (SS), 11 (Mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan SPI *library*.
5. LED: 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin bernilai nilai *HIGH*, LED *on*, ketika pin bernilai *LOW*, LED *off*.

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

1. I2C: A4 (SDA) dan A5 (SCL). Dukungan I2C (TWI) komunikasi menggunakan perpustakaan *Wire*.
2. Aref. Tegangan referensi (0 sampai 5V saja) untuk *input* analog. Digunakan dengan fungsi analogReference ().
3. *Reset*. Bawa baris ini *LOW* untuk *me-reset* mikrokontroler.

2.5.5 Komunikasi

Arduino Uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega328 menyediakan UART TTL (5V) untuk komunikasi serial, yang tersedia di pin digital 0 (RX) dan 1 (TX). Sebuah ATmega8U2 sebagai saluran komunikasi serial melalui USB dan sebagai *port virtual com* untuk perangkat lunak pada komputer. *Firmware* '8 U2 menggunakan *driver* USB standar COM, dan tidak ada *driver* eksternal yang diperlukan. Namun, pada Windows diperlukan, sebuah *file* inf. Perangkat lunak Arduino terdapat monitor serial yang memungkinkan digunakan memonitor data tekstual sederhana yang akan dikirim ke atau dari *board* Arduino. LED RX dan TX di papan tulis akan berkedip ketika data sedang dikirim melalui *chip* USB-*to*-serial dengan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah *Software* Serial *library* memungkinkan untuk berkomunikasi secara serial pada salah satu pin digital pada *board* Uno. ATmega328 juga mendukung I2C (TWI) dan komunikasi SPI. Perangkat lunak Arduino memiliki perpustakaan yang dapat untuk menyederhanakan penggunaan bus I2C, lihat dokumentasi untuk rincian. Untuk komunikasi SPI, menggunakan perpustakaan SPI.

2.6 Web hosting [6]

Web hosting itu sendiri adalah sebuah layanan *internet* yang akan berfungsi untuk menyewakan tempat untuk menyimpan berbagai macam data atau dokumen yang dibutuhkan oleh sebuah *web*. Data-data yang dimaksudkan disini seperti gambar, *email*, aplikasi, *database*, dll. Atau dengan kata lain jika diibaratkan *web hosting* adalah sebuah *mall*, dan *web* Anda diibaratkan sebagai sebuah kios yang menyewa tempat di salah satu bagian *mall* tersebut. Pihak *mall* akan memberikan berbagai macam fasilitas yang dibutuhkan oleh si penyewa ruko atau kios tersebut. Begitu juga dengan *web hosting* yang akan menyewakan tempatnya untuk pengguna *web*.

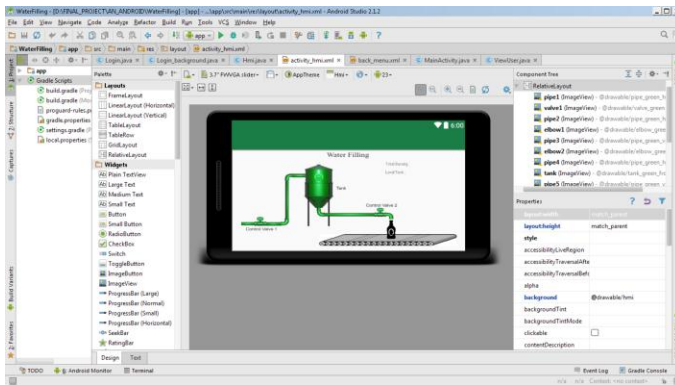
Sedangkan kegunaan *web hosting* itu sendiri adalah untuk mempermudah para pengguna *web* untuk dapat menyimpan data yang diperlukan dalam sebuah *web* yang dikelolanya. Tentunya akan sedikit merepotkan jika tidak menyimpan data atau *file* di *web hosting*. Selain itu sebuah *web hosting* juga dapat menyimpan *email*, dapat menyimpan informasi di *internet*, dapat menyimpan *video*, dapat menyimpan gambar, dapat digunakan untuk membuat blog, dapat membuat *web*, dapat mempublikasikan tulisan, dan juga dapat digunakan untuk membuat survei. Biasanya tempat sebuah *web hosting* menarik bayaran per bulan.

2.7 Software Android Studio [7]

Android Studio merupakan *software* yang digunakan untuk membuat aplikasi Android. Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya:

- Sistem versi berbasis *Gradle* yang fleksibel
- *Emulator* yang cepat dan kaya fitur
- Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
- *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
- *Template* kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
- Alat pengujian dan kerangka kerja yang ekstensif
- Alat *Lint* untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
- Dukungan C++ dan NDK
- Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*

Tampilan dari *software* Android Studio ditunjukkan seperti pada Gambar 2.11.



Gambar 2.11 Tampilan Antarmuka Android Studio

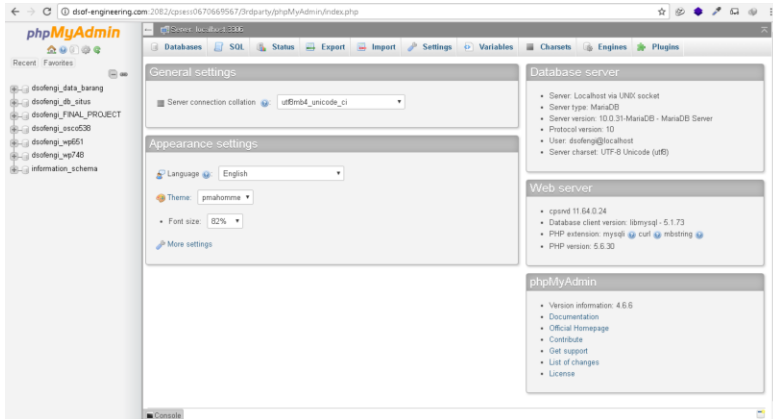
2.8 *phpMyAdmin* [8]

phpMyAdmin adalah sebuah aplikasi/perangkat lunak bebas (*open-source*) yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi *database* MySQL melalui jaringan lokal maupun *internet*. *phpMyAdmin* mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perijinan (*permissions*), dan lain-lain).

Pada dasarnya, mengelola basis data dengan MySQL harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (*command line*) untuk setiap maksud tertentu. Jika seseorang ingin membuat basis data (*database*), ketikkan baris perintah yang sesuai untuk membuat basis data. Jika seseorang menghapus tabel, ketikkan baris perintah yang sesuai untuk menghapus tabel. Hal tersebut tentu saja sangat menyulitkan karena seseorang harus hafal dan mengetikkan perintahnya satu per satu.

Saat ini banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola basis data dalam MySQL, salah satunya adalah *phpMyAdmin*. Dengan *phpMyAdmin*, seseorang dapat membuat *database*, membuat tabel, mengisi data, dan lain-lain dengan mudah, tanpa harus menghafal baris perintahnya.

PhpMyAdmin merupakan bagian untuk mengelola basis data MySQL yang ada di komputer. Tampilan dari *phpMyAdmin* ditunjukkan seperti pada Gambar 2.12.



Gambar 2.12 Tampilan *phpMyAdmin*

2.9 Wemos D1 Mini [9]

Wemos merupakan salah satu arduino *compatible development board* yang dirancang khusus untuk keperluan IoT. *Wemos* menggunakan *chip* SoC WiFi yang cukup terkenal saat ini yaitu ESP8266. Cukup banyak modul WiFi yang menggunakan SoC ESP8266. Namun *Wemos* memiliki beberapa kelebihan tersendiri yang menurut saya sangat cocok digunakan untuk Aplikasi IoT. Bentuk *board* dapat dilihat pada Gambar 2.13



Gambar 2.13 *Wemos D1 Mini*

Beberapa alasan yang membuat *Wemos* cukup menarik untuk di oprek adalah sebagai berikut :

1. *Arduino compatible*, artinya dapat diprogram menggunakan Arduino IDE dengan sintaks program dan *library* yang banyak terdapat di *internet*.
2. *Pinout* yang *compatible* dengan Arduino uno, *Wemos D1 R2* merupakan salah satu *product* yang memiliki bentuk dan *pinout*

standar seperti Arduino Uno. Sehingga memudahkan kita untuk menghubungkan dengan Arduino *Shield* lainnya.

3. *Wemos* dapat *running stand alone* tanpa perlu dihubungkan dengan mikrokontroler. Berbeda dengan modul WiFi lain yang masih membutuhkan mikrokontroler sebagai pengontrol, *Wemos* dapat *running stand alone* karena didalamnya sudah terdapat CPU yang dapat diprogram melalui Serial *port* ataupun via OTA (*Over The Air*) atau transfer program secara *wireless*.
4. *High Frequency CPU*, dengan processor utama 32bit berkecepatan 80MHz *Wemos* dapat mengeksekusi program lebih cepat dibanding dibandingkan mikrokontroler 8 bit yang digunakan di Arduino.
5. Dukungan *High Level Language*, Selain menggunakan Arduino IDE *Wemos* juga dapat diprogram menggunakan bahasa Python dan Lua. Sehingga memudahkan bagi *network programmer* yang belum terbiasa menggunakan Arduino.

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

BAB III

PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai perancangan sistem dari alat yang akan dibuat. Dimana akan dibagi menjadi dua bagian, untuk Rudy Setiawan bertugas untuk merancang dan membuat *hardware* yang berhubungan dengan konveyor, lalu merancang dan membuat *software* HMI untuk PC menggunakan *software* LabVIEW, dan juga merangkai rangkaian elektronika yang mendukung dengan jalannya konveyor. Untuk Miftahul Falahi Alfafa bertugas untuk merancang dan membuat *hardware* yang berhubungan dengan tangki penampungan cairan, lalu merancang dan membuat *software* HMI untuk Android menggunakan *software* Android Studio, dan juga merangkai rangkaian elektronika yang mendukung dengan jalannya sistem pengukuran *level* tangki penampungan

3.1 Blok Diagram Cara Kerja Alat

1. Blok diagram dapat dilihat pada lampiran A.7. untuk *Sensor level*
2. Blok diagram dapat dilihat pada lampiran A.8 untuk *sensor* pendeteksi wadah
3. Diagram fungsional dapat dilihat pada lampiran A.9 untuk *sensor level*
4. Diagram fungsional dapat dilihat pada lampiran A.10 untuk *sensor* pendeteksi wadah

Untuk menjelaskan cara kerja pada Poin 3.1.1 dan Poin 3.1.2 dapat melihat diagram fungsional pada lampiran A9

3.1.1 Cara Kerja Dari Sistem

1. Cairan Cat akan mengalir ke tangki penampungan ketika *control valve* 1 dan pompa 1 aktif, dengan syarat *sensor level* tidak mendeteksi adanya cairan cat
2. *Control Valve* 1 dan pompa 1 tidak aktif ketika *sensor level* sudah memenuhi syarat, atau *volume* cairan cat pada tangki penampungan adalah 100 %.
3. Cairan cat pada tangki penampungan akan dialirkan menuju ke wadah-wadah kecil yang berjalan di atas konveyor.

3.1.2 Cara Kerja Dari Sistem Proses Pengisian Cat

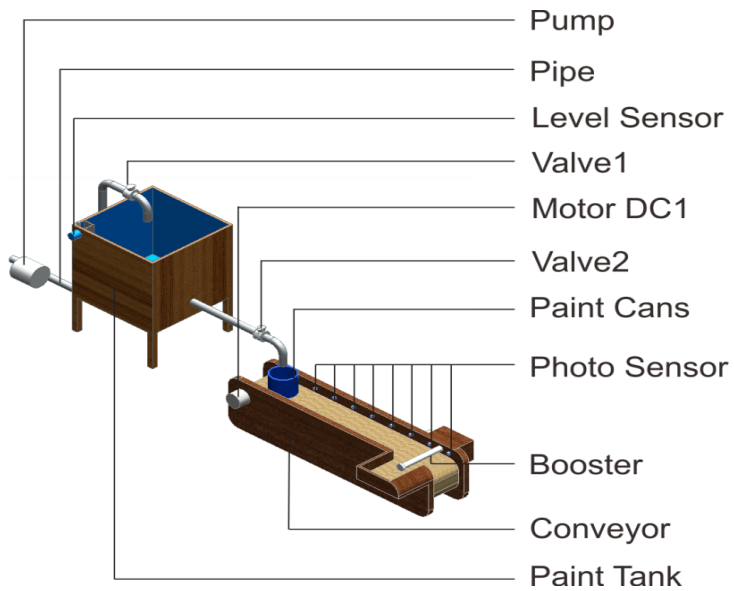
1. Ketika *sensor* wadah 1 pada konveyor mendeteksi wadah maka konveyor akan *on* dan motor konveyor akan berputar.
2. Ketika *sensor* wadah 4 mendeteksi benda maka motor konveyor akan berhenti berputar, lalu *control valve* 2 dan pompa 2 akan *on* selama 5 detik (diasumsikan sebanyak 1 liter).
3. Setelah *control valve* 2 aktif selama 5 detik maka motor konveyor akan berputar kembali.
4. Ketika *sensor* wadah 5 mendeteksi wadah, maka motor konveyor akan berhenti berputar, dan motor pendorong akan *on* selama 3 detik untuk memindahkan wadah yang berisi cairan menuju ke proses pengepakan.

3.2 Perencanaan dan Pembuatan *Hardware*

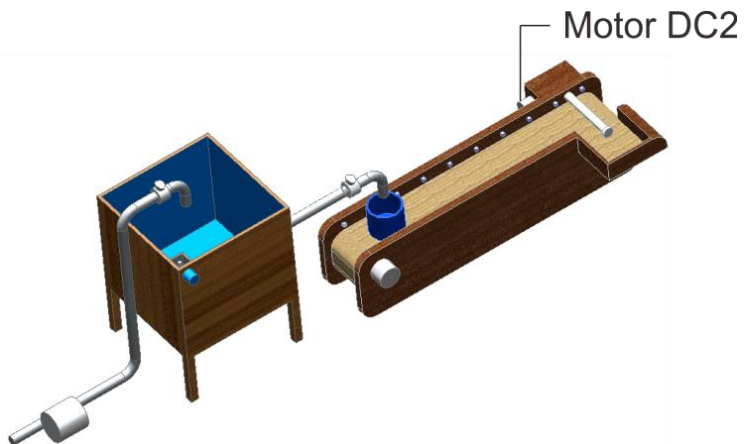
Sesuai dengan permasalahan dan tujuan dari alat kami ini, maka langkah awal kami dalam perencanaan perancangan dan pembuatan alat kami ini adalah dengan mencari literatur dan makalah yang pernah membahas tentang telemetri *level* tangki serta proses pengisian cairan agar didapatkan data yang benar, alat yang digunakan berupa:

1. Tangki penampungan cairan, dengan penanggung jawab adalah Miftahul Falahi Alfafa
2. *Ten Turn* Potentiometer yang dipasangkan pada tangki untuk *sensor level*, dengan penanggung jawab adalah Rudy Setiawan
3. *Photodiode* dan *IR Led* yang dipasangkan pada konveyor sebagai pendeteksi botol, dengan penanggung jawab adalah Rudy Setiawan
4. *Driver relay* untuk kontrol tegangan motor DC pada konveyor, dengan penanggung jawab adalah Rudy Setiawan.
5. *Driver relay* untuk kontrol tegangan motor pompa untuk *mensupply* cairan ke dalam tangki, dengan penanggung jawab adalah Miftahul Falahi Alfafa.
6. *Driver relay* untuk kontrol *solenoid valve* pada buka tutup aliran cairan, dengan penanggung jawab adalah Miftahul Falahi Alfafa.

Berikut adalah desain dari perancangan *hardware* dari alat yang kami buat. Lihat pada Gambar 3.1 dan Gambar 3.2







Gambar 3.1 Rancangan Mekanik Pada Proses Pengisian Cairan

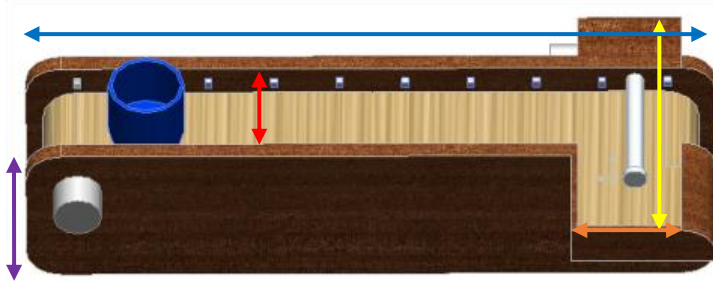


Gambar 3.2 Rancangan Mekanik Pada Proses Pengisian Cairan

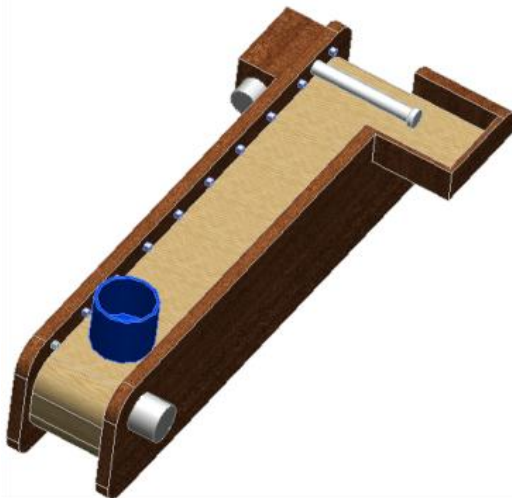
Berikut adalah rancangan mekanik untuk konveyor, dengan penanggung jawab Rudy Setiawan. Dapat dilihat pada Gambar 3.3 dan Gambar 3.4

Dimensi dari konveyor adalah sebagai berikut

- Panjang konveyor 80 cm yang diwakilkan dengan garis 
- Lebar konveyor keseluruhan 24 cm yang diwakilkan dengan garis 
- Lebar *belt* konveyor 8 cm yang diwakilkan garis 
- Tinggi konveyor 24 cm diwakilkan garis 



Gambar 3.3 Rancangan Mekanik Pada Konveyor







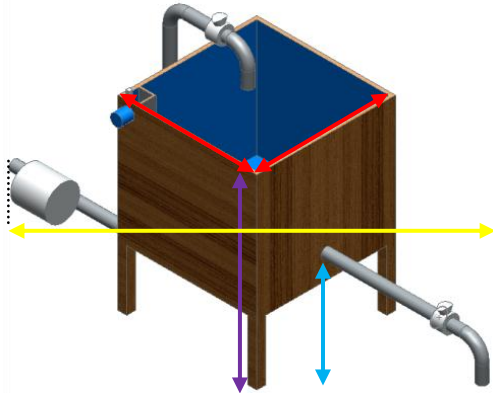
Gambar 3.4 Rancangan Mekanik Pada Konveyor

Berikut adalah rancangan mekanik untuk tangki penampungan, dengan penanggung jawab Miftahul Falahi Alfafa. Dapat dilihat pada Gambar 3.5 dan Gambar 3.6

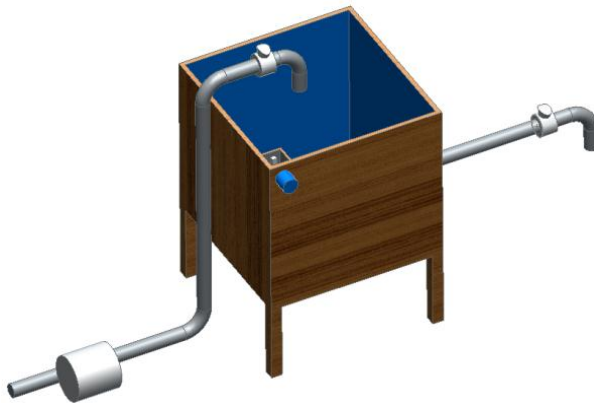
Dimensi dari konveyor adalah sebagai berikut

- tinggi pipa *out* ke tanah 80 cm yang diwakilkan dengan

- garis 
- b. Lebar tangki keseluruhan 65 cm yang diwakilkan dengan garis 
- c. Lebar tangki 17 cm yang diwakilkan garis 
- d. Tinggi tangki 114 cm diwakilkan garis 



Gambar 3.5 Rancangan Mekanik Pada Tangki Penampungan



Gambar 3.6 Rancangan Mekanik Pada Tangki Penampungan

3.3 Perancangan Sistem Arduino

Pada Poin ini menjadi tanggung jawab kedua mahasiswa karena sama-sama menggunakan kontroler Arduino. Sangat penting penggunaan suatu kontroler, yang mana digunakan untuk mengontrol jalannya sistem dari alat kami. Pada tugas akhir ini kami menggunakan

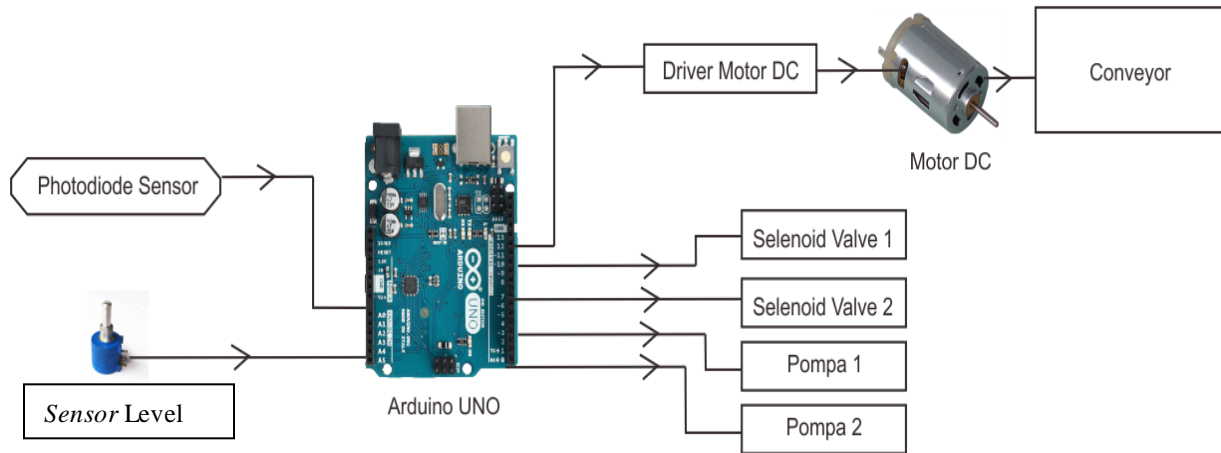
kontroler Arduino UNO. (*board* arduino dapat dilihat pada Gambar 2.10, dan untuk spesifikasi dari Arduino UNO dapat dilihat pada Bab 2 pada sub bagian Poin 2.5)

Berikut ini akan ditampilkakan dari *mapping I/O (Input/Output)* dari Arduino UNO atau bisa disebut *Wiring Diagram I/O* (lihat pada Gambar 3.7). dengan data *mapping* lihat pada Tabel 3.1.

Sebelum menuliskan program pada kontroler Arduino, dibutuhkan *flowchart* untuk alur penulisan program. *Flowchart* dapat dilihat pada Gambar 3.22

Tabel 3.1 *Data Mapping I/O* Arduino UNO

<i>Input / Output</i>	<i>Digital Input</i>	<i>Digital Output</i>	<i>Analog Input</i>
<i>Photodiode Sensor</i>	-----	-----	A0, A1, A2, A3, A5
<i>Ten Turn Potentiometer</i>	-----	-----	A5
<i>Solenoid Valve 1</i>	-----	D6	-----
<i>Solenoid Valve 2</i>	-----	D9	-----
Pompa 1	-----	D7	-----
Pompa 2	-----	D8	-----
Motor Konveyor	-----	D2, D4	-----
Motor Pendorong	-----	D3, D5	-----
<i>Ethernet</i>	D11	D10, D12, D13	-----
<i>Wemos D1</i>	Pin RX (D0)	Pin TX (D1)	-----

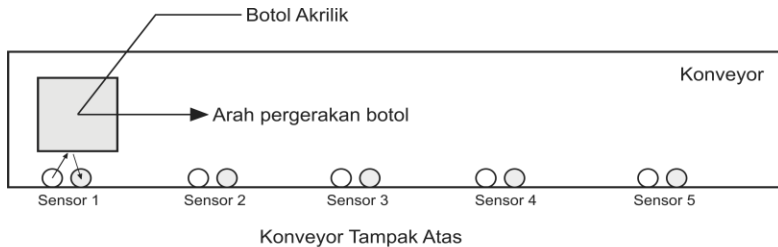


Gambar 3.7 Diagram Blok Sistem Pengisian Cairan Secara Otomatis

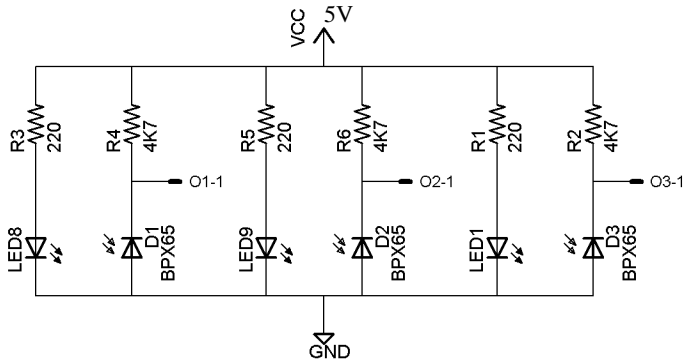
3.4 Perancangan Sistem Sensor Pendeteksi Wadah

Pada Poin ini menjadi tanggung jawab Rudy Setiawan, karena *sensor photodiode* merupakan bagian alat yang mendukung jalannya sistem konveyor. Untuk mendeteksi wadah yang berada diatas konveyor maka dibutuhkanlah sebuah alat yang menggunakan *sensor photodiode*. *Sensor* ini tidak hanya digunakan untuk mendeteksi adanya wadah saja, namun *sensor* ini jika mendeteksi wadah maka akan sekaligus mengirimkan *signal* ke arduino. Yang mana data tersebut akan diolah dan dikirimkan ke *web hosting* melalui perangkat Modul Wemos. Dari data tersebut diolah juga untuk *software* LabVIEW. Dan data dari *web hosting* (data *web hosting* sudah di-upload otomatis ketika *sensor* mendeteksi wadah) digunakan untuk *software* aplikasi HMI Android. Berikut adalah perancangan pemasangan *sensor photodiode* pada konveyor lihat pada Gambar 3.8. Dan untuk rangkaian elektroniknya dapat dilihat pada Gambar 3.9 dan Gambar 3.10.

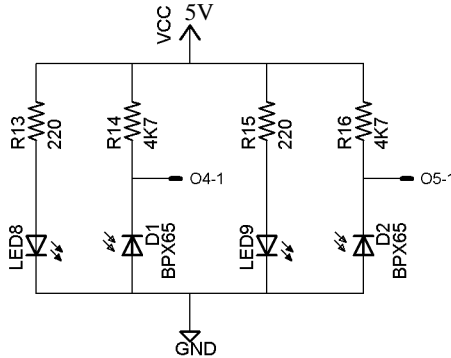
Sebelum menuliskan program pada *software* Arduino maka diperlukan *flowchart*, yang digunakan untuk petunjuk alur penulisan program. *Flowchart* dapat dilihat pada Gambar 3.23



Gambar 3.8 Perancangan Pemasangan *Photodiode Sensor*



Gambar 3.9 Rangkaian Elektronika *Photodiode Sensor* Bagian Pertama

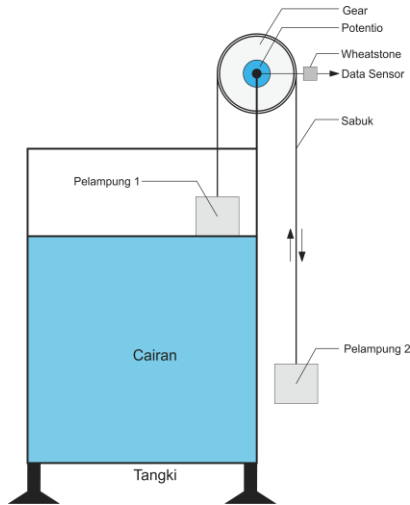


Gambar 3.10 Rangkaian Elektronika *Photodiode Sensor* Bagian Kedua

3.5 Perancangan Sistem *Sensor Level*

Pada Poin ini menjadi tanggung jawab Miftahul Falahi Alfafa, karena *sensor level* adalah alat yang menunjang jalannya sistem tangki penampungan. Pada tugas akhir ini kami menggunakan sebuah tangki penampungan cairan, yang mana cairan ini akan dipindahkan ke dalam wadah-wadah kecil. Namun disini diperluakan sebuah alat untuk mendeteksi *level* caian dalam tangki. Untuk itu kami menggunakan *sensor ten turn potentiometer* untuk mendeteksi *level* cairan dalam tangki. Mengapa kami menggunakan *sensor* ini, dikarenakan putaran dari *sensor* ini sangat panjang sampai 10 kali putaran, dan perubahan data sangat halus dan teliti. Keadaan dalam tangki penampungan harus tetap terjaga *level*nya, agar *supply* ke wadah kecil tidak kekurangan. Berikut adalah rancangan dari pemasangan *sensor Ten Turn*

Potentiometer pada tangki penampungan lihat pada Gambar 3.11. Untuk jalannya program diperlukan alur dalam pembuatannya, maka diperlukan *flowchart*. (dapat dilihat pada Gambar 3.24)



Gambar 3.11 Perancangan *Sensor Ten Turn Potentiometer* di Tangki Penampungan

3.5.1 Rangkaian *Wheatstone Bridge* Dan *ASC*

Untuk membuat kedua rangkaian ini dibutuhkan komponen yang sesuai agar hasil tegangan yang keluar dari kedua rangkaian ini sesuai harapan. Berikut adalah data tegangan yang diinginkan dari rangkaian *Wheatstone* dapat dilihat pada Tabel 3.2. Dan data tegangan yang diinginkan dari rangkaian *Analog Signal Conditioning (ASC)* dapat dilihat pada Tabel 3.3.

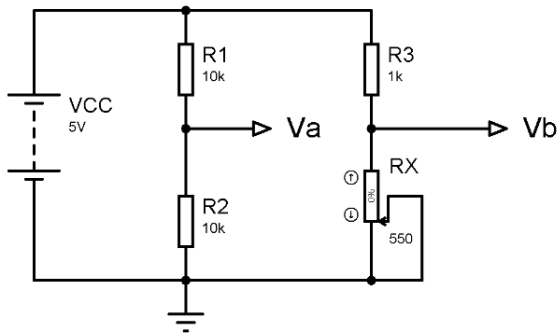
Tabel 3.2 Data *Wheatstone Bridge* Yang Diinginkan

Data Dari <i>Wheatstone Bridge</i>	
Tegangan Minimal	0 Volt
Tegangan Maksimal	2,5 Volt

Tabel 3.3 Data *Analog Signal Conditioning* Yang Diinginkan

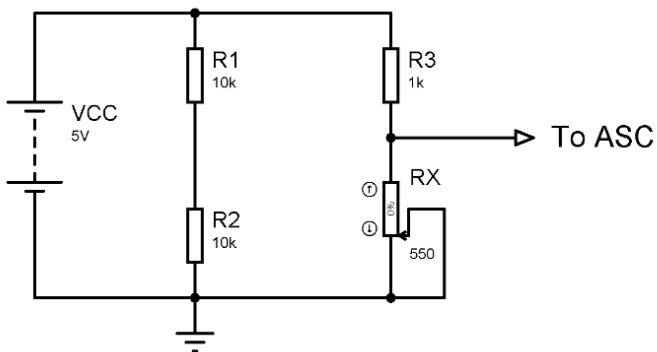
Data dari <i>Analog Signal Conditioning</i>	
Tegangan Minimal	0 Volt
Tegangan Maksimal	5 Volt

Pada Gambar 3.12 adalah rangkaian *wheatstone bridge* dimana ada titik V_a dan V_b untuk melihat beda potensial setelah adanya perubahan nilai R_x (*sensor ten turn potentiometer*)



Gambar 3.12 Rangkaian *Wheatstone* Untuk Mengetahui Beda Potensial

Pada Gambar 3.13 adalah rangkaian *wheatstone bridge* dimana ditunjukkan pin untuk menuju rangkaian ASC



Gambar 3.13 Rangkaian *Wheatstone* Dengan Pin TO ASC

pada Gambar 3.14 adalah rangkaian ASC, dimana rangkaian ini mendapat *input* dari rangkaian *Wheatstone*. Rangkaian ini digunakan untuk merubah dari *input Wheatstone 0 Volt -2,5 Volt* menjadi *0 Volt-5 Volt*.

3.5.2 Perhitungan Rangkaian *Wheatstone Bridge* dan ASC

Perhitungan Rangkaian *Wheatstone*

Berdasarkan Gambar rangkaian *Wheatstone* (lihat pada Gambar 3.12) maka didapatkan Persamaan rumus dan nilai dari V_a , V_b (lihat Persamaan 3.1, 3.2, 3.3)

Ketika R_x bernilai maksimal (550 Ohm) maka :

$$V_a = \left(\frac{R_2}{R_1 + R_2} \right) \times V_{cc} \dots \dots \dots (3.1)$$

$$V_a = \left(\frac{10k}{10k + 10k} \right) \times 5 \dots \dots \dots (3.1)$$

$$V_a = 2,5 \text{ Volt} \dots \dots \dots (3.1)$$

$$V_b = \left(\frac{R_x}{R_x + R_3} \right) \times V_{cc} \dots \dots \dots (3.2)$$

$$V_b = \left(\frac{550}{550 + 1000} \right) \times 5 \dots \dots \dots (3.2)$$

$$V_b = 1,774 \text{ Volt} \dots \dots \dots (3.2)$$

Maka :

$$V_{out} = V_a - V_b = 0,725 \text{ Volt} \dots \dots \dots (3.3)$$

Ketika R_x bernilai minimal (0 Ohm) maka :

Berdasarkan dari Persamaan 3.1 dan Persamaan 3.2, maka didapatkan nilai V_a 2,5 Volt dan V_b 0 Volt. Lalu berdasarkan Persamaan 3.3 maka didapatkan nilai V_{out} 2,5 Volt

Perhitungan Rangkaian *Analog Signal Conditioning*

Berdasarkan Gambar rangkaian *analog signal conditioning* (lihat pada Gambar 3.14) maka didapatkan persamaan rumus dan nilai dari V_a , V_b (lihat Persamaan 3.4, 3.5, 3.6)

$$m = \frac{R4}{R3} \dots\dots\dots (3.4)$$

$$m = \frac{(Vout(max) - Vout(min))}{(Vin(max) - Vin(min))} \dots\dots\dots \beta.4)$$

$$m = \frac{(5 - 0)}{(2,5 - 0)} \dots\dots\dots (3.4)$$

$$m = \frac{5}{2,5} \dots\dots\dots (3.4)$$

,

Sehingga nilai

$$R4 = 5K\Omega$$

$$R3 = 2,5K\Omega$$

$y = mx + b$ karena $Vout(min) = 0$ dan $Vin(min) = 0$ maka :

$$y = mx + b \Leftrightarrow Vout(min) = m * Vin(min) + b \dots\dots\dots \beta.5)$$

$$0 = m * 0 + b \dots\dots\dots \beta.5)$$

$$b = 0 \dots\dots\dots \beta.5)$$

$$b = \left(\frac{R4}{Ros} \right) * V \dots\dots\dots (3.5)$$

Karena nilai

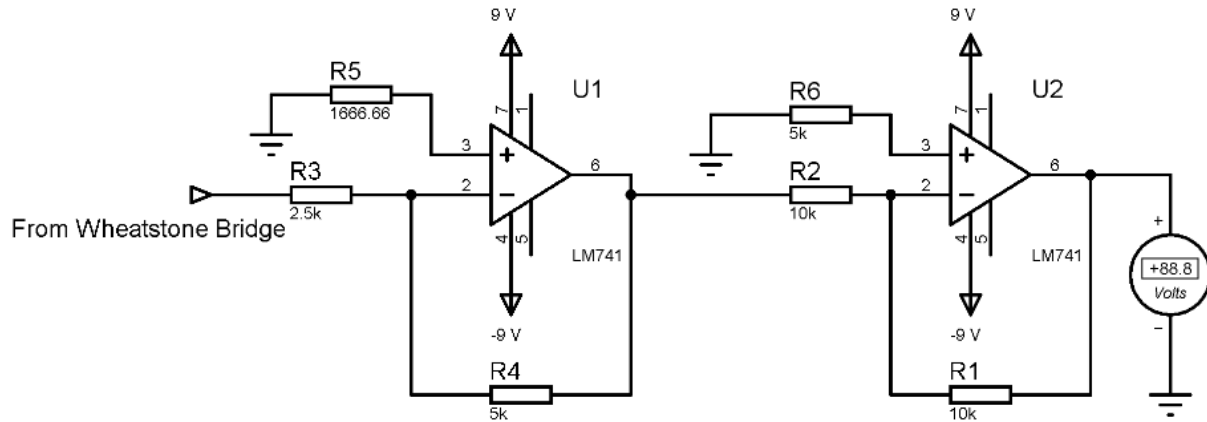
$$Vout(min) = 0$$

maka

$$Ros = 0$$

$$Rcomp = Rf || Ri || Ros \dots\dots\dots (3.6)$$

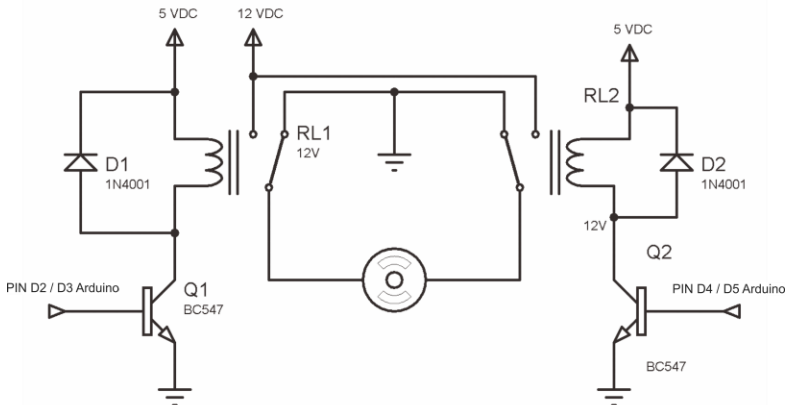
$$Rcomp = 5 || 2,5 || 0 = 1666,6\Omega \dots\dots\dots \beta.6)$$



Gambar 3.14 Rangkaian *Analog Signal Conditioning*

3.6 Perancangan Sistem Kendali Motor DC

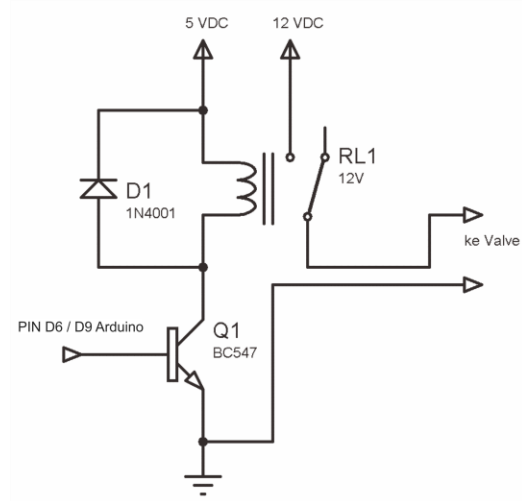
Pada Poin ini menjadi tanggung jawab Rudy Setiawan, dikarenakan motor DC menunjang jalannya sistem konveyor. Motor DC pada konveyor terdapat dua bagian, bagian pertama berfungsi sebagai penggerak *belt* konveyor. Dan bagian kedua adalah sebagai pendorong, maksud dari pendorong adalah memindahkan wadah yang tadinya ada di jalur konveyor untuk dikeluarkan menuju titik luar dari *belt*. Motor pendorong bekerja setelah wadah sudah terisi penuh pada proses pengisian. Rangkaian elektronika dari motor DC dapat dilihat pada Gambar 3.15. Alur Program dapat dilihat pada Gambar 3.27, dan Gambar 3.28.



Gambar 3.15 Rangkaian *Driver* Motor DC

3.7 Perancangan Sistem *Solenoid Valve*

Pada Poin ini menjadi tanggung jawab Miftahul Falahi alfafa. *Solenoid valve* dibagi menjadi dua bagian. Bagian pertama adalah pada titik pengisian cairan pada tangki penampungan, fungsi dari *solenoid valve* ini sebagai penutup atau pembuka aliran cairan menuju tangki penampungan. Dan untuk bagian kedua terletak pada titik pengisian cairan ke wadah yang berada diatas konveyor. Fungsi diberikan *solenoid valve* untuk membuka atau menutup aliran cairan menuju wadah. Rangkaian Elektronika untuk *solenoid valve* dapat dilihat pada Gambar 3.16. Alur program dapat dilihat pada Gambar 3.29 dan Gambar 3.30.

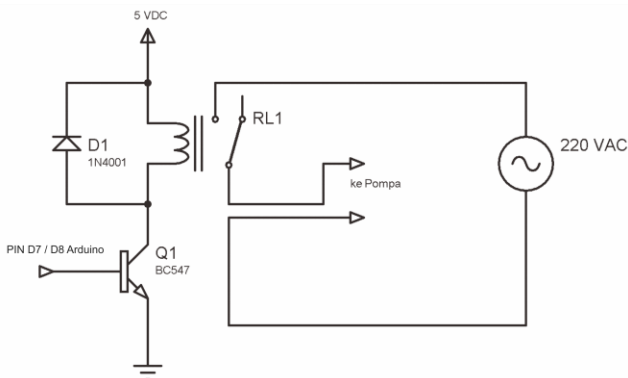


Gambar 3.16 Rangkaian Elektronika *Selenoid Valve*

3.8 Perancangan Sistem Pompa

Pada Poin ini adalah tanggung jawab dari Miftahul Falahi Alfa. Poin ini membahas mengenai pompa. Pompa disini menggunakan tegangan AC 220 Volt, oleh sebab itu diperlukan *driver* untuk mengontrol kapan pompa hidup atau mati. Pompa ini digunakan untuk mensupply cairan menuju tangki utama dan ke wadah yang melintas diatas konveyor.

Rangkaian elektronika dari pompa dapat dilihat pada Gambar 3.17



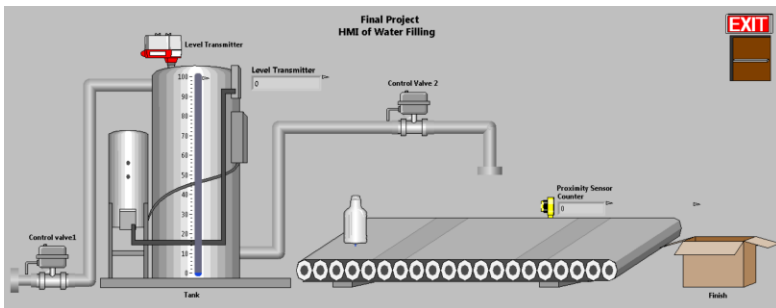
Gambar 3.17 Rangkaian Elektronika Pompa AC 220 Volt

3.9 Perencanaan dan Pembuatan *Software*

Supaya keseluruhan sistem dapat bekerja secara optimal maka diperlukan *software* yang mendukung kinerja dari sistem. Untuk itu dibuatlah *software* HMI (*Human Machine Interface*) untuk PC (*Personal Computer*) dan *Android Phone*. Untuk keseluruhan blok fungsional sistem seperti ditunjukkan pada Gambar 3.19.

3.9.1 HMI (*Human Machine Interface*) Untuk PC

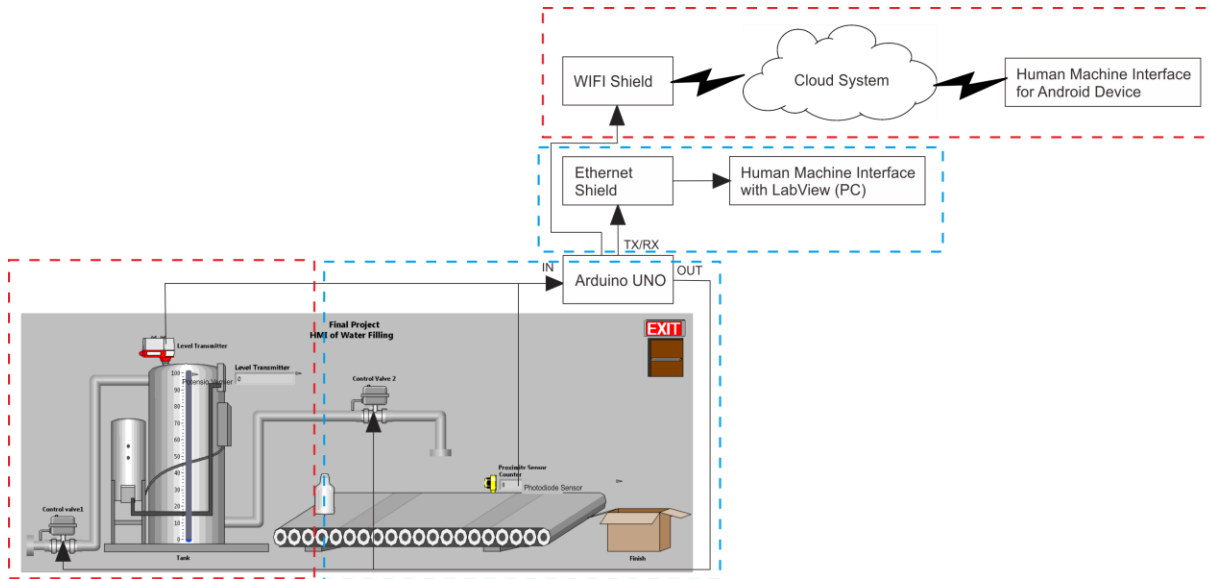
Untuk memudahkan pengamatan dan pemantauan dari kinerja alat maka dibuatlah HMI untuk PC yang dibuat dengan menggunakan *software* LabVIEW. Komunikasi antara HMI LabVIEW dengan *hardware* adalah dengan menggunakan modul *ethernet shield*. Tampilan dari *software* HMI untuk PC adalah seperti ditunjukkan pada Gambar 3.18. dengan penanggung jawab Rudy Setiawan.



Gambar 3.18 HMI (*Human Machine Interface*) Untuk PC (*Personal Computer*)

3.9.2 HMI (*Human Machine Interface*) Untuk **Android**

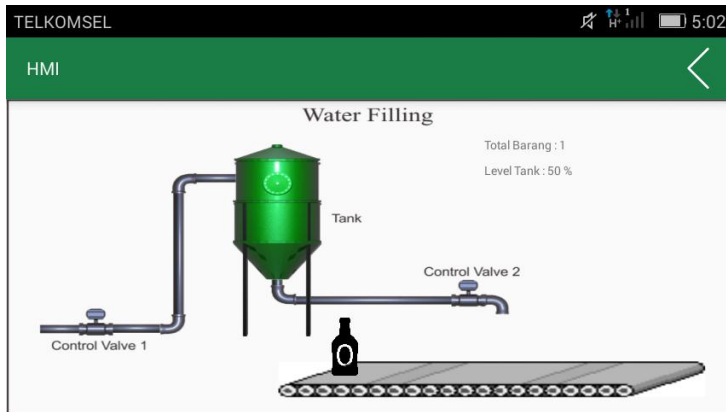
Untuk memudahkan orang yang bepekerjaan dalam memantau kinerja dari *hardware* maka dibuatlah aplikasi HMI untuk *Android Phone*. *Software* yang digunakan untuk membuat aplikasi HMI untuk *Android* adalah *Android Studio 2.1.2*. Media komunikasi yang digunakan antara aplikasi HMI dengan *hardware* adalah *WIFI* dengan menggunakan modul *ESP8266 wifi shield*. Tampilan dari aplikasi HMI untuk *Android Phone* seperti ditunjukkan pada Gambar 3.20. dengan penanggung jawab Miftahul Falahi Alfafa.



Keterangan :

- - - - - Rudy Setiawan
- - - - - Miftahul Falahi Alfafa

Gambar 3.19 Blok Fungsional Sistem



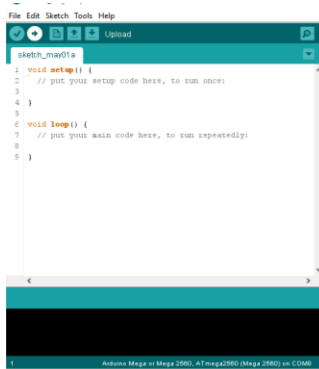
Gambar 3.20 HMI (*Human Machine Interface*) Untuk *Android Phone*

3.9.3 Perancangan *Software* Arduino

Arduino menggunakan keluarga mikrokontroler ATmega yang dirilis oleh Atmel sebagai basis, namun ada individu/perusahaan yang membuat *clone* Arduino dengan menggunakan mikrokontroler lain dan tetap kompatibel dengan Arduino pada *level hardware*. Untuk fleksibilitas, program dimasukkan melalui *bootloader* meskipun ada opsi untuk *mem-bypass bootloader* dan menggunakan *downloader* untuk memprogram mikrokontroler secara langsung melalui *port ISP*.







Pengontrol yang digunakan pada alat ini adalah Arduino ATmega328, dipilihnya *chip* ini karena fitur – fitur yang dimiliki cukup lengkap dan *simple*, selain itu juga memiliki kecepatan yang lebih baik yaitu satu siklus mesin untuk satu instruksi dengan kecepatan hingga 16 MHz serta banyak tersedia dipasaran. Untuk dapat bekerja, mikrokontroler perlu mendapat tegangan kerja sebesar 9 Volt dengan arus 40 mA dan *ground* serta *clock*, dengan *clock* yang digunakan pada perancangan ini sebesar 16 MHz.

Pada pembuatan perangkat lunak ini menggunakan *software* Arduino dengan tampilan Gambar 3.21.



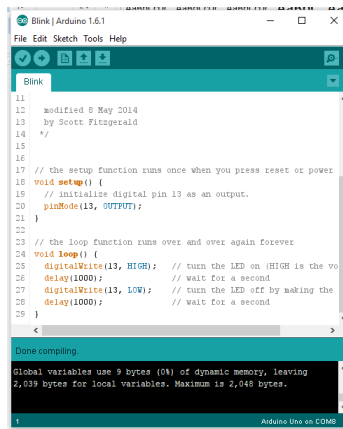
Gambar 3.21 Tampilan *Software* Arduino

Dari Gambar 3.21 di atas adalah *software* Arduino yang akan digunakan untuk membuat program pada tugas akhir ini. Sebelum menjelaskan langkah-langkah pembuatan program pada tugas akhir ini, maka terlebih dahulu dijelaskan beberapa ikon yang sering di gunakan, yaitu :

1.  Merupakan ikon *create new project*. Ikon ini berfungsi untuk memulai sebuah projek program.
2.  Adalah Ikon Menu *Verify* yang bergambar ceklis, ini berfungsi untuk mengecek program yang ditulis apakah ada yang salah atau *error*.
3.  Adalah Ikon Menu *Upload* yang bergambar panah ke arah kanan, ini berfungsi untuk memuat atau mentransfer program yang dibuat di *software* Arduino ke *hardware* Arduino.
4.  Adalah Ikon Menu *Open* yang bergambar panah ke arah atas, ini berfungsi untuk membuka program yang disimpan atau membuka program yang sudah dibuat dari pabrikan *software* Arduino.
5.  Adalah ikon *save* yang bergambar panah ke arah bawah, ini berfungsi untuk menyimpan program yang telah dibuat atau dimodifikasi
6.  Adalah Ikon Menu *Serial Monitor* yang bergambar kaca pembesar (*loop*), ini berfungsi mengirim atau menampilkan serial komunikasi data saat dikirim dari *hardware* Arduino.

Setelah mengetahui beberapa ikon yang digunakan dan telah membuat program dalam *software*, maka langkah selanjutnya yang harus dilakukan adalah cara meng-*upload* program pada Arduino. Untuk meng-*upload* program pada Arduino terdapat langkah-langkah yang harus dilakukan. Berikut langkah-langkah meng-*upload* program pada Arduino:

1. Langkah pertama yang dilakukan setelah membuat program adalah melakukan *verify* program yang digunakan dengan cara menekan tombol *verify*. Jika tidak ada *error* maka akan muncul kata *done compiling* seperti Gambar 3.22 di bawah ini.



```
11
12   modified 8 May 2014
13   by Scott Fitzgerald
14   */
15
16
17 // the setup function runs once when you press reset or power
18 void setup() {
19   // initialize digital pin 13 as an output.
20   pinMode(13, OUTPUT);
21 }
22
23 // the loop function runs over and over again forever
24 void loop() {
25   digitalWrite(13, HIGH); // turn the LED on (HIGH is the vo
26   delay(1000);           // wait for a second
27   digitalWrite(13, LOW); // turn the LED off by making the
28   delay(1000);          // wait for a second
29 }
```

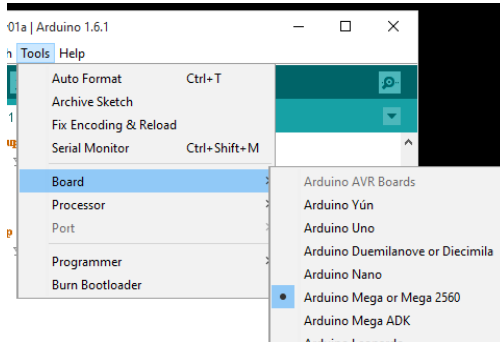
Done compiling.

Global variables use 9 bytes (0% of dynamic memory, leaving 2,039 bytes for local variables. Maximum is 2,048 bytes.

Arduino Uno on COM6

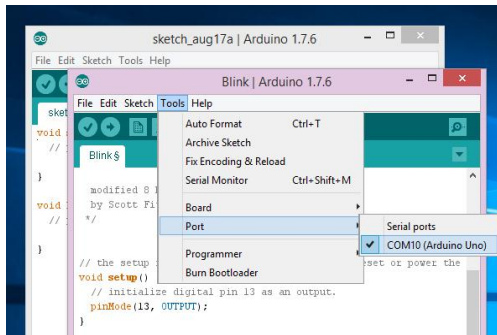
Gambar 3.22 Tampilan Arduino Ketika Tidak Terjadi *Error* Pada Saat *Verify* Program

2. Langkah kedua adalah menyambungkan *hardware* Arduino dengan laptop atau pc yang digunakan.
3. Langkah ketiga adalah memilih *board* Arduino yang digunakan dengan cara menekan *Tools* lalu *board* dan *board* yang digunakan seperti pada Gambar 3.23



Gambar 3.23 Memilih *Board* Arduino Pada *Software* Arduino IDE

4. Langkah keempat adalah memilih *port* yang digunakan pada laptop atau PC seperti pada Gambar 3.24 di bawah ini.

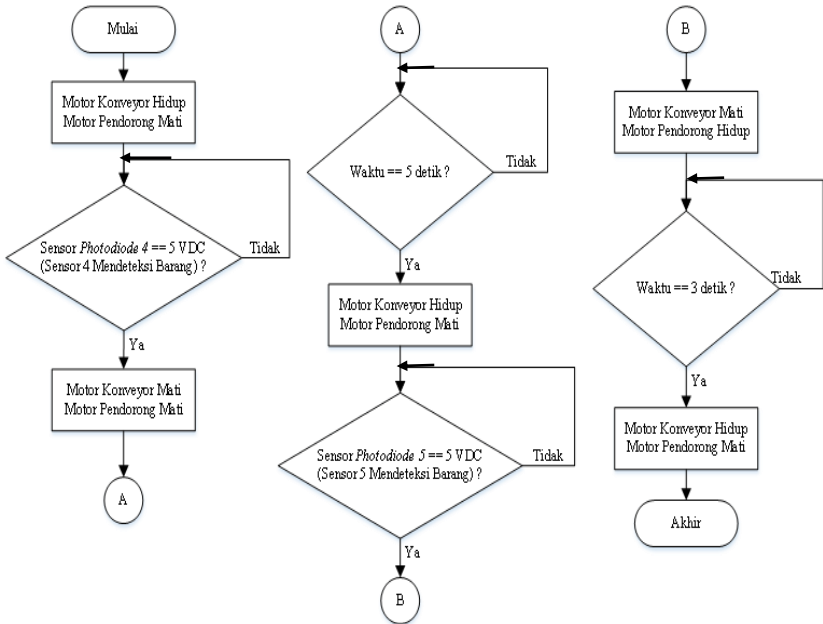


Gambar 3.24 Memilih *Port* Pada Arduino

5. Langkah terakhir adalah menekan ikon *upload* atau menekan *file* → *upload*.

3.9.4 Perancangan *Software Sensor Photodiode*

Pada Poin ini adalah tanggung jawab dari Rudy Setiawan. Pada *Point* ini adalah program yang akan di *upload* ke Arduino, yang mana akan membuat *sensor photodiode* bekerja sebagaimana mestinya. Berikut adalah alur program untuk *sensor photodiode* (lihat pada Gambar 3.25).



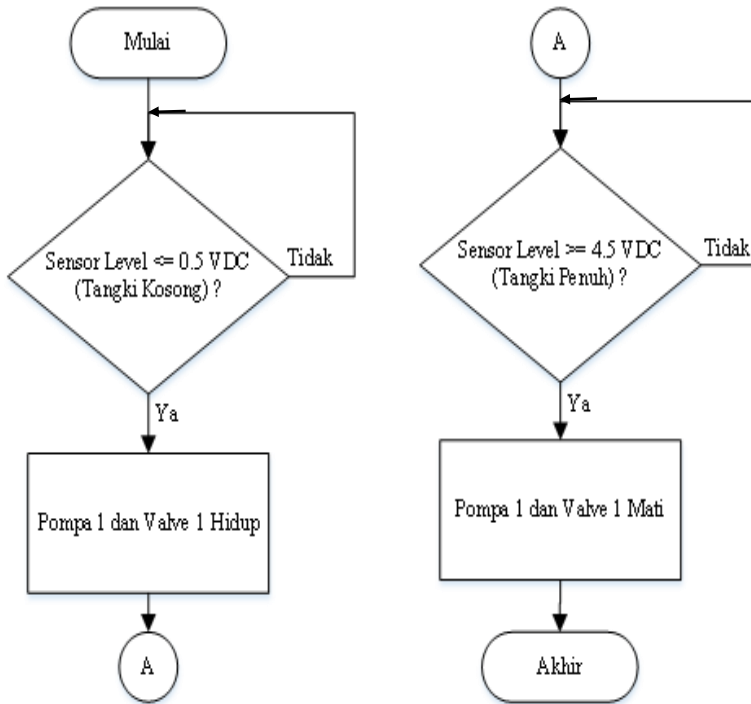
Gambar 3.25 Flowchart Sensor Photodiode

3.9.5 Perancangan Software Sensor Level

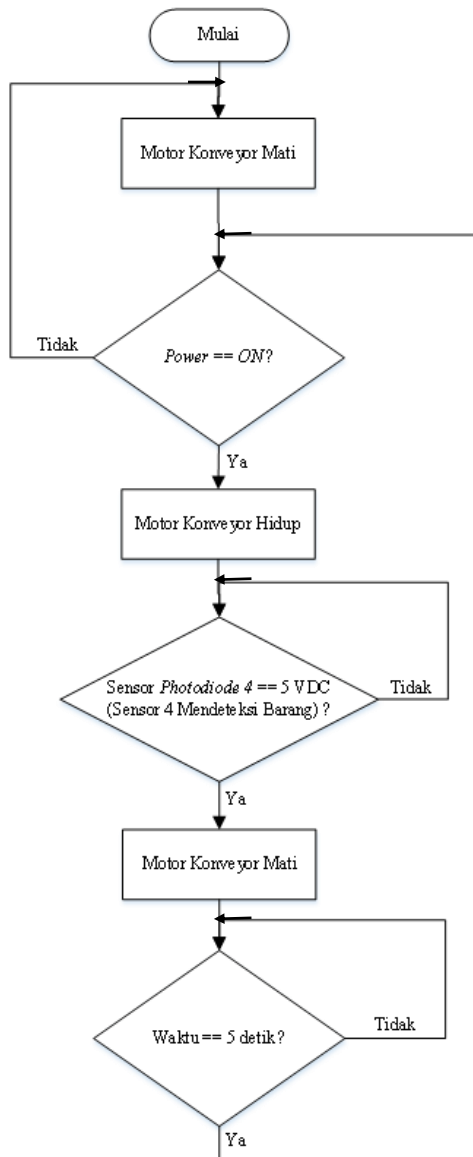
Pada Poin ini adalah tanggung jawab dari Miftahul Falahi Alfafa. Pada Poin ini adalah program yang akan di-upload ke arduino agar *sensor level* dapat berfungsi sebagaimana mestinya. Berikut adalah alur program untuk *sensor level* (lihat pada Gambar 3.26).

3.9.6 Perancangan Software Motor DC

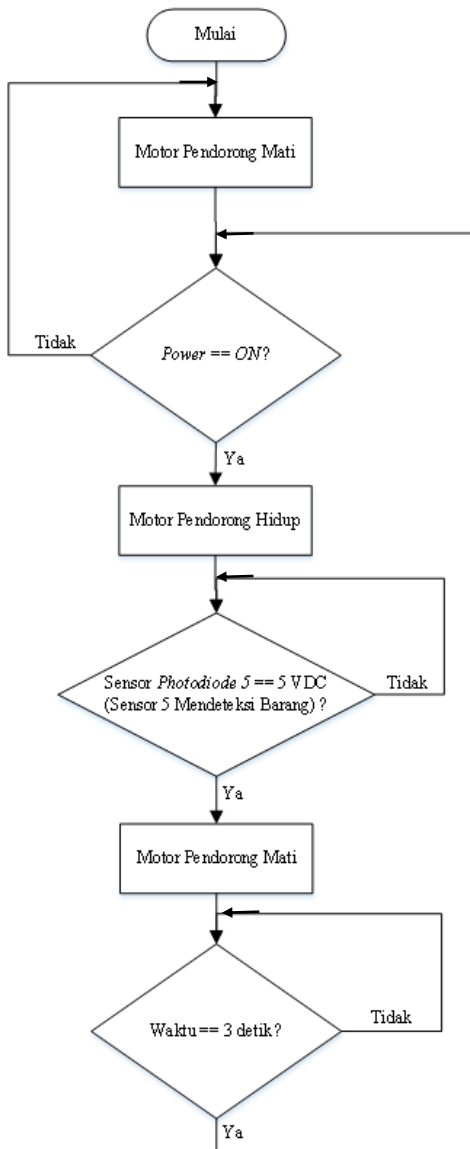
Pada Poin ini adalah tanggung jawab Rudy Setiawan. Poin ini adalah menjelaskan tentang program dari Motor DC yang mana digunakan untuk penggerak konveyor dan pendorong wadah. Untuk itu maka diperlukan program yang akan di-upload ke Arduino. Berikut adalah alur program dari motor pendorong konveyor (lihat pada Gambar 3.27), dan untuk motor pendorong (lihat pada Gambar 3.28)



Gambar 3.26 *Flowchart Untuk Sensor Level*



Gambar 3.27 Flowchart Motor Konveyor



Gambar 3.28 Flowchart Motor Pendorong

3.9.7 Perancangan Software Selenoid Valve dan Pompa

Pada Poin ini adalah tanggung jawab Miftahul Falahi Alfafa. Poin ini menjelaskan tentang *selenoid valve* dan pompa, yang mana kedua komponen ini digunakan sebagai pen-supply cairan kedalam tangki penampung maupun ke wadah. Untuk itu diperlukan sebuah program untuk mengatur jalannya dari kedua komponen ini. Agar sesuai dengan sebagaimana mestinya maka diperlukan alur program yang sesuai.

Berikut adalah alur program untuk *selenoid valve* dan pompa untuk tangki penampungan (lihat pada Gambar 3.29), dan *selenoid valve* dan pompa untuk wadah (lihat pada Gambar 3.30)

3.9.8 Perancangan Software Shield Ethernet Arduino

Pada Poin ini adalah tanggung jawab dari Rudy Setiawan. Pada tugas akhir ini kami membuat sistem monitoring atau biasa disebut HMI. *Shield Ethernet* ini adalah sebuah modul yang digunakan untuk pengiriman data ke HMI. Berikut adalah alur program pengiriman data dari modul *ethernet* menuju HMI. (Lihat pada Gambar 3.31).

3.9.9 Perancangan Software Wemos D1 Mini

Pada Poin ini adalah tanggung jawab dari Miftahul Falahi Alfafa. Poin ini adalah program yang akan di *upload* ke modul *Wemos D1 Mini* yang mana modul ini bertugas untuk mengupload data ke *web hosting*, data ini digunakan untuk pemrosesan data untuk HMI Android. Berikut adalah alur program untuk modul *Wemos D1 Mini*. (lihat pada Gambar 3.32)

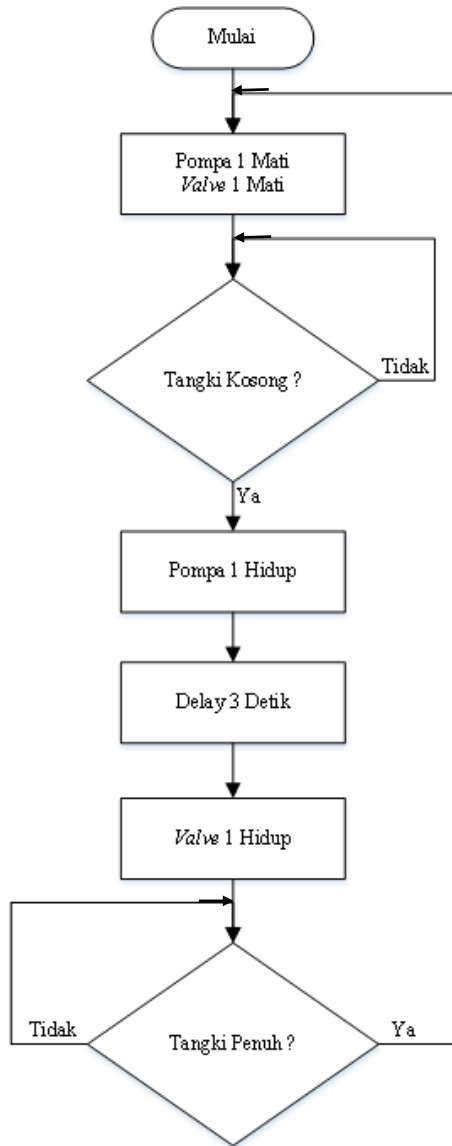
3.9.10 Perancangan Website

1. Tampilan *Sub Domain* *waterfilling.ds-of-engineering.com*

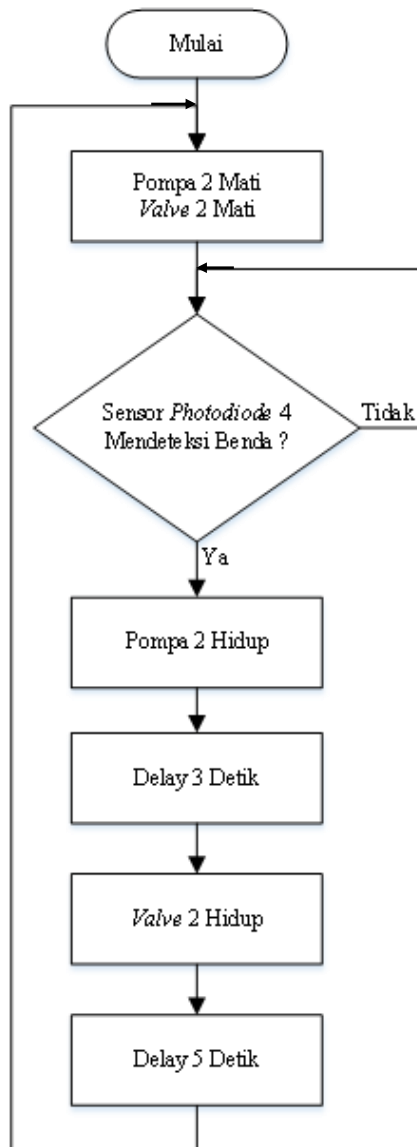
Sub domain pada Gambar 3.33 digunakan untuk melihat data dan mengupdate data yang akan ditampilkan pada HMI android.

2. Tampilan Data *Web hosting*

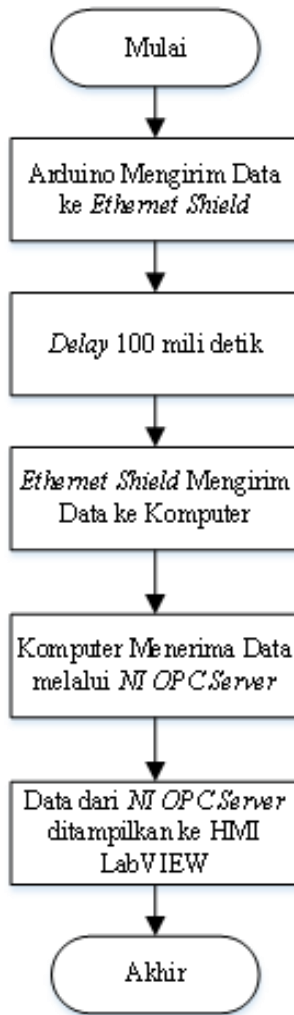
Pada Gambar 3.34 merupakan data data yang akan dikirim ke aplikasi HMI Android. Data-data tersebut perlu diparsing terlebih dahulu untuk mendapatkan nilai yang sebenarnya sehingga dapat dimasukkan pada HMI Android.



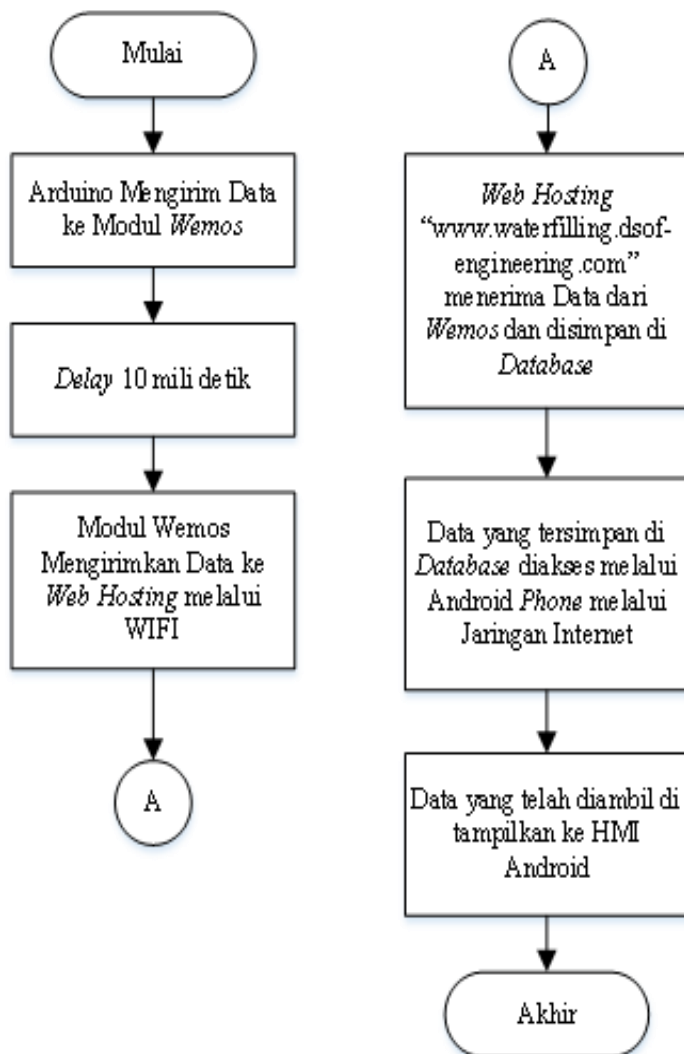
Gambar 3.29 Flowchart Selenoid Valve dan Pompa Tangki Utama



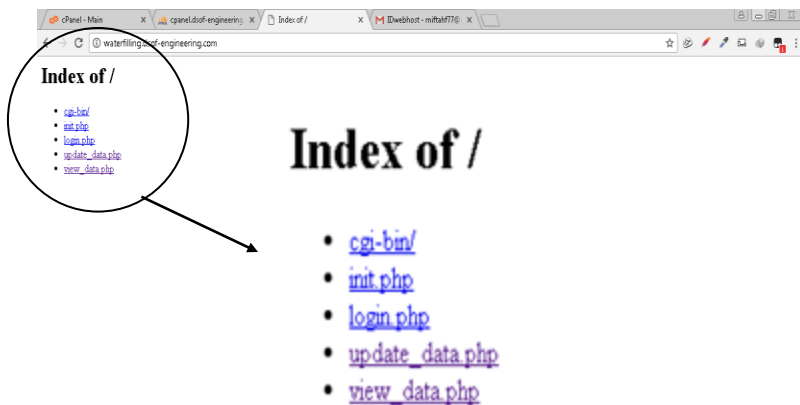
Gambar 3.30 Flowchart Selenoid Valve dan Pompa Wadah



Gambar 3.31 Flowchart Shield Ethernet



Gambar 3.32 Flowchart Wemos D1 Mini



Gambar 3.33 Tampilan Subdomain



\$00#0\$11#1\$25#2\$30#3\$40#4\$50#5\$60#6\$70#7\$80#8\$90#9\$100#10\$110#11\$120#12\$130#13\$14137#14

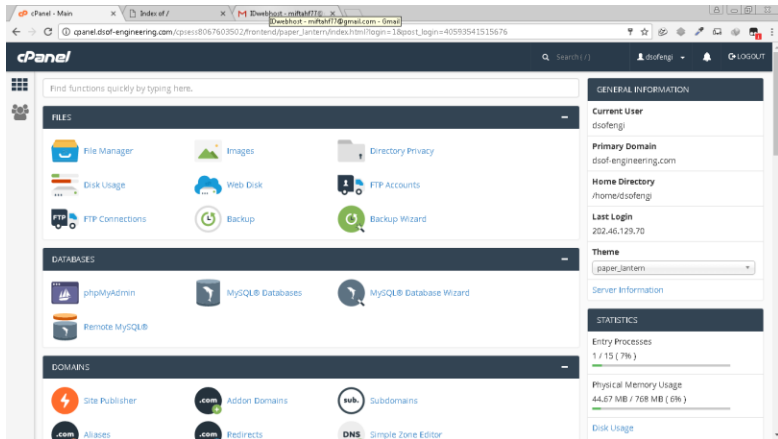
Gambar 3.34 Tampilan Data Web hosting

3.9.11 Perancangan Sistem Web hosting

Pada Poin ini adalah tanggung jawab dari Miftahul Falahi Alfafa.

1. Kontrol Panel Website dsdf-engineering.com

Kontrol panel (cPanel) adalah *online* kontrol panel yang dapat digunakan untuk mengatur *website*, membuat *email* atau banyak hal lainnya seperti instalasi *script*. Pada Gambar 3.35 adalah kontrol panel untuk menyimpan data yang berhubungan dengan HMI Android.



Gambar 3.35 Kontrol Panel *Website*

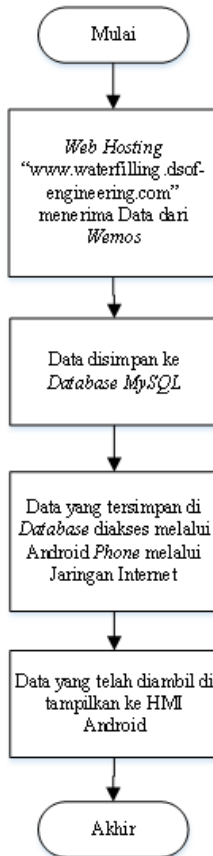
Web hosting berfungsi sebagai tempat penyimpanan data dari data yang telah dikirim oleh modul *Wemos D1 Mini*. Data yang sudah disimpan lalu diakses melalui jaringan *internet* secara *global*, maksudnya data dapat diakses dari mana saja asalkan mempunyai jaringan *internet*. Berikut adalah alur kerja dari *web hosting* (lihat pada Gambar 3.36).

3.9.12 Perancangan Sistem *phpMyAdmin*

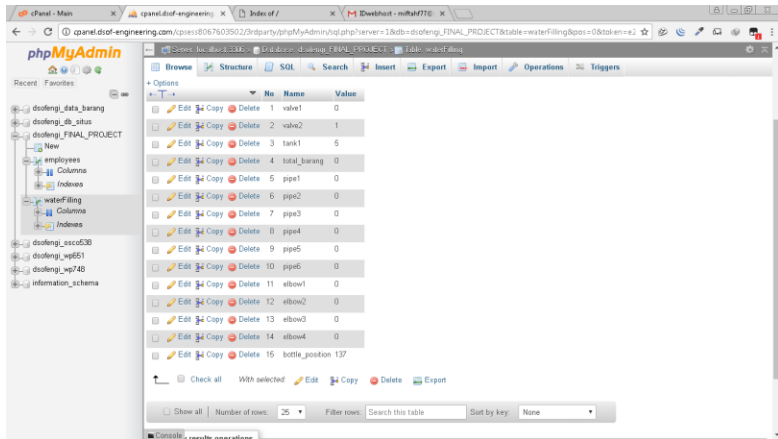
Pada Poin ini adalah tanggung jawab dari Miftahul Falahi Alfafa. Poin ini menjelaskan tentang *phpMyAdmin*, yang mana bertugas sebagai alat atau *aplikasi* pengelola *database*. Data yang berasal dari *sensor* yang telah dikirim oleh modul *Wemos D1 Mini*. Data akan disimpan.

1. Tampilan *phpMyAdmin*

phpMyAdmin adalah aplikasi yang digunakan untuk mengelola *database* pada suatu *website*. Pada gambar 3.37 merupakan tampilan data data yang akan digunakan pada HMI Android



Gambar 3.36 Flowchart Web hosting



Gambar 3.37 Tampilan Database phpMyAdmin

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

BAB IV HASIL SIMULASI DAN UJI COBA

Bab ini akan dijelaskan mengenai uji coba pada masing-masing komponen dan juga data dari komponen. Pada bab ini Rudy Setiawan bertanggung jawab pada pengujian yang berhubungan dengan konveyor dari *sensor* maupun HMI terdapat pada Poin 4.2, 4.4, 4.7, 4.8. Begitu juga dengan Miftahul Falahi Alfafa bertanggung jawab pada pengujian yang berhubungan dengan Tangki penampungan dari *sensor* maupun HMI terdapat pada Poin 4.1, 4.3,4.5, 4.6, 4.9, 4.10. Tiap Poin dijelaskan siapa yang bertanggung jawab.

4.1 Pengujian *Sensor level*

Pada Poin ini yang bertanggung jawab adalah Miftahul Falahi Alfafa. Berikut akan dijelaskan cara pengukuran *level* dari *sensor* potensiometer sekaligus hasil data yang diperoleh dari pengukuran *sensor* potensiometer. Untuk ilustrasi pengukuran ditunjukkan seperti pada Gambar.3.11

Sabuk akan bergerak naik atau turun tergantung dari *level* air didalam tangki. Pergerakan sabuk akan mempengaruhi nilai resistansi dari potensiometer yang dikonversi menjadi tegangan dengan menggunakan rangkaian jembatan *Wheatstone* yang selanjutnya masuk ke *input* kontroler untuk mengukur *level* air didalam tangki.

4.1.1 Data Hasil Pengukuran

Pada Tabel 4.1 adalah data tegangan dari keluaran rangkaian *Wheatstone*.

Tabel 4.1 Data Hasil Pengukuran Pada *Output Wheatstone Bridge*

Data Dari <i>Wheatstone Bridge</i>	
Tegangan Minimal	0,10 Volt
Tegangan Maksimal	2,54 Volt

Tabel 4.2 Data Hasil Pengukuran Pada *Output Analog Signal Conditioning*

Data dari <i>Analog Signal Conditioning</i>	
Tegangan Minimal	0,56 Volt
Tegangan Maksimal	4,22 Volt

Pada Tabel 4.2 adalah data tegangan yang keluar dari rangkaian ASC.

Pengukuran / pengujian pada *sensor level* dilakukan dengan cara mengukur tegangan *output* dari rangkaian *Analog Signal Conditioning (ASC)* lihat pada Gambar 3.14 dengan menggunakan *Voltmeter* dari *level* air pada tangki sebesar 0% sampai *level* pada tangki mencapai 100%. Data yang didapat dari pengukuran *level* air ditunjukkan seperti pada Tabel 4.3.

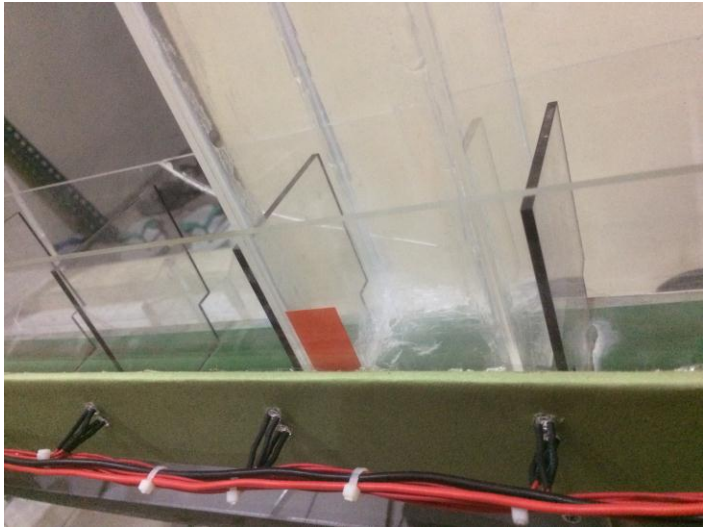
Tabel 4.3 Data *Sensor* Potensiometer Untuk Pengukuran *Level* Air

No	<i>Level</i> (%)	Tinggi Air (cm)	Tegangan terukur (Volt)
1	0	0	0,08
2	10	2,85	0,2
3	20	5,7	2,5
4	30	8,55	3,3
5	40	11,4	3,7
6	50	14,25	3,9
7	60	17,1	4
8	70	19,95	4,18
9	80	22,8	4,27
10	90	25,65	4,35
11	100	28,5	4,45

Total *volume* tangki yang digunakan untuk menyimpan cairan adalah 6412,5 cm³.

4.2 Pengujian *Sensor Photodiode*

Pada Poin ini yang bertanggung jawab adalah Rudy Setiawan. Berikut ini akan dijelaskan mengenai *sensor photodiode*, *sensor* ini digunakan untuk mendeteksi wadah yang melintas diatas konveyor. (lihat pada Gambar 4.1 dan Gambar 4.2)



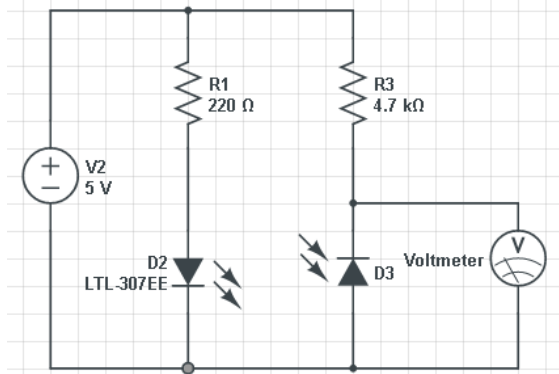
Gambar 4.1 *Sensor Photodiode* Dengan Ada Wadah di Konveyor



Gambar 4.2 *Sensor Photodiode* Dengan Tidak Ada Wadah di Konveyor

Untuk model peletakan *sensor photodiode* secara lebih jelas ditunjukkan seperti pada Gambar 3.8. Dan rangkaian untuk menguji *sensor level* ditunjukkan pada Gambar 4.3

Dari gambar ilustrasi pada Gambar 3.8 maka didapatkan data sesnsor *photodiode* seperti pada Tabel 4.4, Tabel 4.5 dan Tabel 4.6.



Gambar 4.3 Rangkaian Untuk Pengujian *Sensor* Pendeteksi Wadah

Untuk melakukan pengujian dilakukan dengan cara meletakkan botol akrilik di depan *sensor photodiode*, ketika terdapat benda di depan *sensor photodiode* maka nilai tegangan *output* pada rangkaian *sensor photodiode* akan berubah (berkurang nilai tegangannya / mendekati 0 Volt). Ketika nilai tegangan *output photodiode* mendekati 0 Volt maka ditulis YA pada data tabel percobaan. Dan ketika nilai *output photodiode* tetap tinggi / jauh dari nol Volt maka ditulis TIDAK pada data Tabel 4.6, 4.7, 4.8.

Tabel 4.4 Data *Sensor Photodiode* Untuk Botol Akrilik Tanpa Air

Perc.	<i>Sensor 1</i>	<i>Sensor 2</i>	<i>Sensor 3</i>	<i>Sensor 4</i>	<i>Sensor 5</i>
1	Ya	Ya	Ya	Ya	Ya
2	Ya	Tidak	Tidak	Ya	Ya
3	Ya	Ya	Tidak	Ya	Ya
4	Ya	Tidak	Ya	Ya	Ya
5	Ya	Ya	Ya	Ya	Ya

Keterangan :

YA = *Sensor photodiode* mendeteksi benda.

TIDAK = *Sensor photodiode* tidak mendeteksi benda.

Tabel 4.5 Data *Sensor Photodiode* Untuk Botol Akrilik Berisi Air

Perc.	<i>Sensor 1</i>	<i>Sensor 2</i>	<i>Sensor 3</i>	<i>Sensor 4</i>	<i>Sensor 5</i>
1	Ya	Ya	Ya	Ya	Ya
2	Ya	Ya	Tidak	Ya	Ya
3	Ya	Ya	Ya	Ya	Ya
4	Ya	Tidak	Ya	Ya	Ya
5	Ya	Ya	Ya	Ya	Ya

Tabel 4.6 Data *Sensor Photodiode* Untuk Botol Akrilik Yang Dilapisi Stiker Warna Orange

Perc.	<i>Sensor 1</i>	<i>Sensor 2</i>	<i>Sensor 3</i>	<i>Sensor 4</i>	<i>Sensor 5</i>
1	Ya	Ya	Ya	Ya	Ya
2	Ya	Ya	Ya	Ya	Ya
3	Ya	Ya	Ya	Ya	Ya
4	Ya	Ya	Ya	Ya	Ya
5	Ya	Ya	Ya	Ya	Ya

Keterangan :

Ya = *Sensor photodiode* mendeteksi barang yang lewat

Tidak = *Sensor photodiode* tidak mendeteksi barang yang lewat

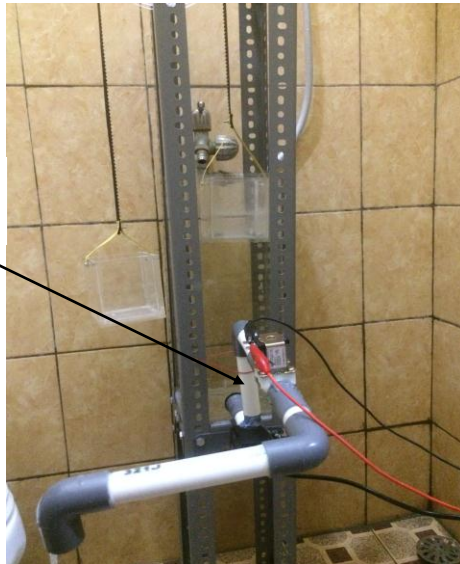
Dari data *sensor photodiode* yang telah diperoleh seperti pada Tabel 4.4, 4.5 dan 4.6, maka botol akrilik yang telah dilapisi stiker warna orange lebih mudah terdeteksi oleh *sensor photodiode*, oleh karena itu dipilih botol yang telah dilapisi oleh skotlet warna orange sebagai wadah untuk pengisian cairan di conveyor.

4.3 Pengujian *Solenoid Valve*

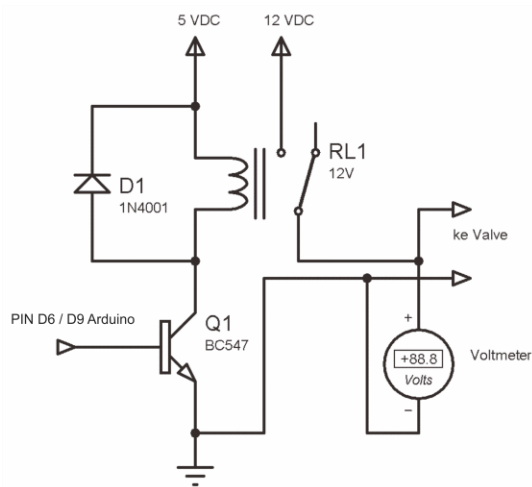
Pada Poin ini yang bertanggung jawab adalah Miftahul Falahi Alfa. *Solenoid valve* ini berfungsi seperti stop kran air, namun digerakkan secara otomatis menggunakan tegangan DC (lihat pada

Gambar 4.4). Data pengujian *solenoid valve* dapat dilihat pada Tabel 4.7. Lalu untuk rangkaian pengujian dapat dilihat pada Gambar 4.5

Solenoid Valve



Gambar 4.4 *Solenoid Valve*



Gambar 4.5 Rangkaian Untuk Pengujian *Solenoid Valve*

Pengujian pada *solenoid valve* dilakukan dengan cara memberikan tegangan pada pin *Input* (ke Pin Arduino / pada transistor Q1), kemudian pada bagian *output* rangkaian (ke Valve) diukur dengan menggunakan *Voltmeter*. Sehingga didapatkan data seperti pada Tabel 4.7.

Tabel 4.7 Pengujian *Solenoid Valve*

No	Pin D6/D9 Arduino ,(Volt)	<i>Voltmeter</i> (Volt)	Keadaan <i>Valve</i>
1	0	0	Tertutup
2	4,8	11,8	Terbuka

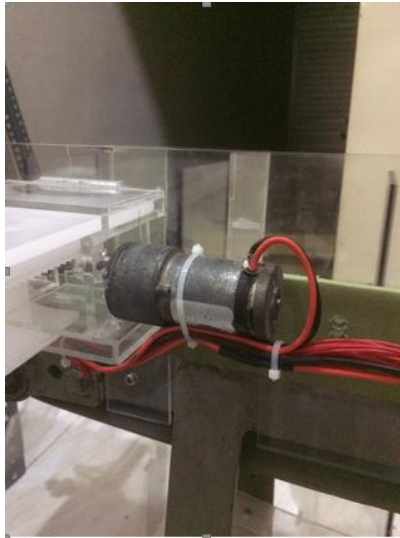
4.4 Pengujian Motor DC

Pada Poin ini yang bertanggung jawab adalah Rudy Setiawan. Penggunaan motor DC disini sebagai penggerak konveyor, dan juga sebagai pendorong wadah menuju tempat pengepakan. (lihat Gambar 4.6 dan Gambar 4.7)

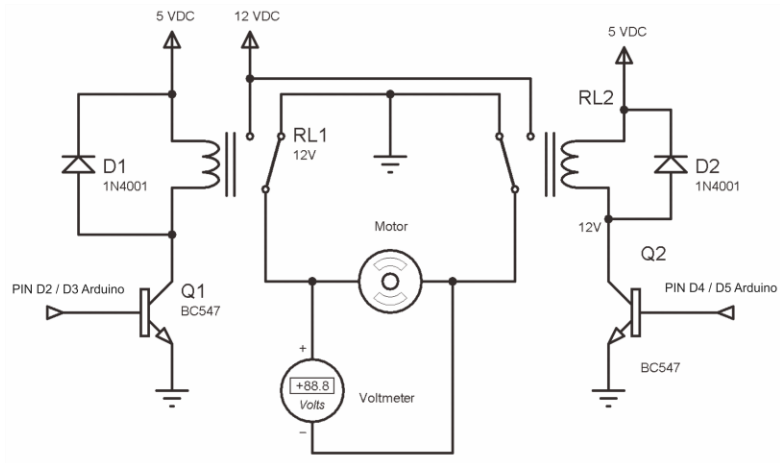
Rangkaian pengujian dapat dilihat pada Gambar 4.8 dan hasil pengujian dapat dilihat pada Tabel 4.8



Gambar 4.6 Motor DC Sebagai Penggerak Konveyor



Gambar 4.7 Motor DC Sebagai Pendorong Wadah



Gambar 4.8 Rangkaian Untuk Pengujian Motor DC

Pengujian pada Motor DC dilakukan dengan memasang motor DC pada rangkaian *relay* sehingga motor DC dapat berputar searah jarum jam atau berlawanan arah jarum jam. Kemudian pada bagian *input* rangkaian (ke Pin Arduino pada transistor Q1 dan Q2) diberi tegangan

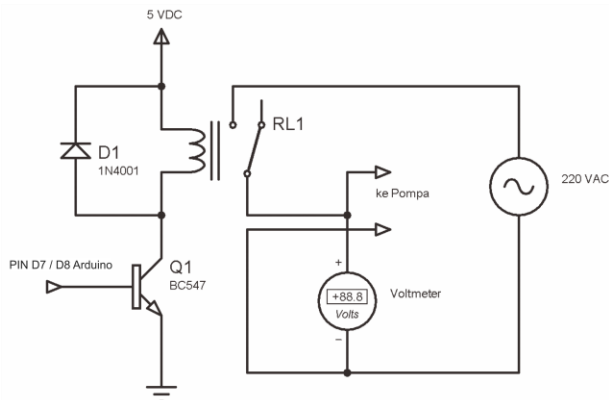
sebesar 5 Volt dari Arduino sehingga motor DC berputar searah jarum jam atau berlawanan arah jarum jam. Kemudian diukur berapa tagangan pada bagian *output* rangkaian *relay* untuk mengetahui berapa tegangan yang dihasilkan dari rangkaian tersebut. Sehingga didapat hasil percobaan seperti pada Tabel 4.8

Tabel 4.8 Pengujian Motor DC

No	Pin D2/D3 Arduino (Volt)	Pin D4/D5 Arduino (Volt)	Voltmeter (Volt)	Arah Putar Motor DC
1	0	0	0	Diam
2	0	4,8	11,9	Kanan
3	4,8	0	-11,9	Kiri
4	4,8	4,8	0	Diam

4.5 Pengujian Pompa

Pada Poin ini yang bertanggung jawab adalah Miftahul Falahi Alfafa dan Rudy Setiawan. Pompa diperlukan untuk *men-supply* cairan ke dalam tangki penampungan maupun ke wadah. Untuk itu diperlukan pengujian agar sistem berjalan lancar. Rangkaian untuk pengujian pompa dapat dilihat pada Gambar 4.9 dan hasil pengujian dapat dilihat pada Tabel 4.9



Gambar 4.9 Rangkaian Pengujian Untuk Pompa

Pada pengujian pompa dilakukan dengan memasang pompa ke rangkaian *driver* pompa. Kemudian rangkaian *driver* diberi tegangan pada bagian *input* (ke Pin Arduino / pada transistor Q1), kemudian diukur tegangan *output* pada rangkaian *driver*, sehingga didapat data hasil pengujian seperti pada Tabel 4.9.

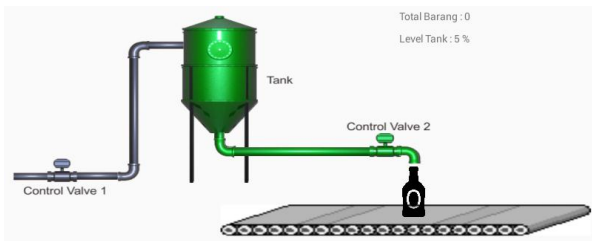
Tabel 4.9 Hasil Pengujian Pompa

No	Pin D7/D8 Arduino (Volt)	Voltmeter (Volt)	Keadaan Pompa
1	0	0 VAC	Mati
2	4,8	220 VAC	Aktif

4.6 Pengujian HMI Android

Pada Poin ini yang bertanggung jawab adalah Miftahul Falahi Alfafa. Berikut akan ditampilkan hasil pengujian dari konektivitas dari HMI Android.

Pengujian pada HMI Android dilakukan dengan cara menjalankan aplikasi HMI dan dilihat / dipantau proses penerimaan data dari *hardware* kemudian dihitung kecepatan penerimaan data dari aplikasi tersebut atau seberapa besar tundaan (*delay*) yang dihasilkan selama proses pengiriman. Hasil Pengujian dapat dilihat pada Tabel 4.10.



Gambar 4.10 Tampilan HMI Android

Konektivitas ke *server* dan penerimaan data dikatakan OK jika nilai-nilai pada aplikasi HMI Android pada Gambar 4.10 (seperti nilai Total Barang, Level Tank, Control Valve 1, Control Valve 2 dan Posisi botol) berubah sesuai dengan data yang ada pada *Web hosting* seperti pada Gambar 4.15.

Tabel 4.10 Pengujian Konektivitas HMI Android

<i>Run</i>	Konektivitas ke <i>Server</i>	Penerimaan Data	<i>Delay</i> (Detik)	Kecepatan <i>Running App</i>	<i>Error</i>
1	OK	OK	1	Normal	No
2	OK	OK	2	Normal	No
3	OK	OK	1	Normal	No
4	OK	OK	0,5	Normal	No
5	OK	OK	1	Normal	No
6	OK	OK	2	Normal	No
7	OK	OK	1	Normal	No
8	OK	OK	2	Normal	No
9	OK	OK	1	Normal	No
10	OK	OK	1	Normal	No

4.7 Pengujian HMI LabVIEW

Pada Poin ini yang bertanggung jawab adalah Rudy Setiawan. Berikut akan ditampilkan hasil pengujian konektivitas dari HMI LabVIEW. Hasil pengujian dapat dilihat di Tabel 4.11.

Pada pengujian HMI untuk PC (*Personal Computer*) dilakukan dengan cara menjalankan aplikasi HMI LabVIEW kemudian dihubungkan ke *Ethernet Shield* dengan menggunakan kabel *Ethernet*, kemudian aplikasi HMI tersebut dijalankan. Setelah itu dilakukan proses pengiriman data dari Arduino ke HMI LabVIEW dan dihitung tundaan (*delay*) dari proses pengiriman. Data hasil percobaan didapatkan seperti pada Tabel 4.11.

Tabel 4.11 Pengujian Konektivitas HMI LabVIEW

<i>Run</i>	Konektivitas ke Modul <i>Ethernet Shield</i>	Penerimaan Data	<i>Delay</i> (Detik)	Kecepatan <i>Running App</i>	<i>Error</i>
1	OK	OK	1	Normal	No
2	OK	OK	2	Normal	No
3	OK	OK	1	Normal	No

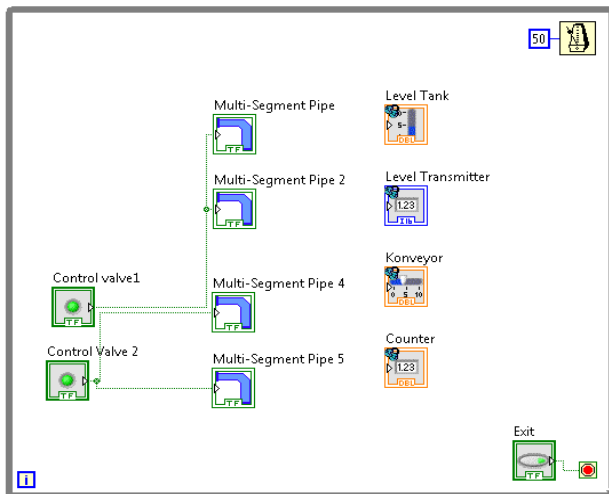
<i>Run</i>	Konektivitas ke Modul <i>Ethernet Shield</i>	Penerimaan Data	<i>Delay</i> (Detik)	Kecepatan <i>Running App</i>	<i>Error</i>
4	OK	OK	0,5	Normal	No
5	OK	OK	1	Normal	No
6	OK	OK	2	Normal	No
7	OK	OK	1	Normal	No
8	BAD	BAD	5	Normal	Yes
9	OK	OK	1	Normal	No
10	BAD	BAD	7	Normal	Yes

Keterangan :

OK : proses pengiriman data berhasil.

BAD : proses pengiriman data gagal.

Normal : Aplikasi HMI LabVIEW berjalan di komputer secara normal



Gambar 4.11 Diagram Blok HMI LabVIEW

Item ID	Data Type	Value	Timestamp	Quality	Update Count
Water Filling.Arduino1.control_valve1	Boolean	Unknown	13:14:28.281	Bad	1
Water Filling.Arduino1.control_valve2	Boolean	Unknown	13:14:28.281	Bad	1
Water Filling.Arduino1.conveyor	Word	Unknown	13:14:37.292	Bad	1
Water Filling.Arduino1.counter	Word	Unknown	13:14:37.292	Bad	1
Water Filling.Arduino1.full indicator	Boolean	Unknown	13:14:28.281	Bad	1
Water Filling.Arduino1.tank_level	Word	Unknown	13:14:37.292	Bad	1

Gambar 4.12 Tampilan *OPC Client* Untuk Mengetahui Status Pengiriman Data Pada Modul *Shield Ethernet*

Konektivitas HMI LabVIEW dikatakan OK jika HMI LabVIEW dapat menerima data dari *Ethernet Shield* atau nilai dari kolom *Value* pada Gambar 4.12 sesuai dengan nilai *variable* pada Diagram Blok LabVIEW seperti pada Gambar 4.11.

4.8 Pengujian *Shield Ethernet*

Pada Poin ini yang bertanggung jawab adalah Rudy Setiawan. Berikut adalah hasil dari pengujian *shield ethernet*. Pengujian pada *hardware Shield Ethernet* dilakukan dengan cara mengirimkan data dari *Shield Ethernet* ke HMI LabVIEW melalui kabel *Ethernet* dan dilakukan perhitungan tundaan (*delay*) dari proses pengiriman tersebut. Hasil dari pengujian dapat dilihat pada Tabel 4.12

Tabel 4.12 Pengujian *Shield Ethernet*

Data	Status Pengiriman (<i>Ethernet Shield</i> ke Komputer)	<i>Delay</i> (Detik)
1	Sukses	2
2	Sukses	3
3	Sukses	2
4	Sukses	4
5	Sukses	5

Keterangan : Menggunakan Kabel *Ethernet* Standar

```

if(tank_level>0 && fl==0){
    Mb.R[2]=tank_level;
    delay(10);
    sprintf(kirim, "%i:%i:%i:%i:%i:", valve1_status, valve2_status, tank_level, count, bottle_pos);
    Serial.println(kirim);
    delay(1000);
}

```

Gambar 4.13 Kode Untuk Mengirimkan Nilai *Sensor Level* ke HMI LabVIEW

Status pengiriman dikatakan sukses jika data yang dikirimkan oleh modul *Shield Ethernet* seperti pada Gambar 4.13 dapat diterima oleh *OPC Client*. Sehingga nilai pada kolom *Value* pada Gambar 4.12 sama dengan nilai yang dikirim oleh modul *Shield Ethernet*.

4.9 Pengujian Wemos D1 Mini

Pada Poin ini yang bertanggung jawab adalah Miftahul Falahi Alfafa. Berikut adalah hasil pengujian modul *Wemos D1 Mini*. Pengujian Modul *Wemos D1 Mini* dilakukan dengan cara mencoba untuk melakukan proses pengiriman data dari modul *Wemos* ke *Web hosting* melalui jaringan WIFI. Kemudian dihitung tundaan (*delay*) dari proses pengiriman tersebut. Hasil pengujian dapat dilihat pada Tabel 4.13

Tabel 4.13 Pengujian Modul *Wemos D1 Mini*

Data	Status Pengiriman (<i>Wemos</i> ke <i>Web hosting</i>)	<i>Delay</i> (Detik)
1	Sukses	1
2	Sukses	1
3	Sukses	1
4	Sukses	2
5	Sukses	1

```

if (client.connect(host, 80)) {
  //Serial.println("connected");
  String url = "/konveyor/add?";
  String url = "/update_data.php?";
  url += "valve1=";
  url += valve1;
  url += "&valve2=";
  url += valve2;
  url += "&tank_level=";
  url += tank_level;
  url += "&total_barang=";
  url += total_barang;
  url += "&bottle_pos=";
  url += bottle_pos;

  url += "waktu=";
  url += waktu;
  url += "&jumlah=";
  url += jumlah;
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");
  //Serial.println("oke");
}

```

Gambar 4.14 Kode Untuk Mengirimkan Data dari Modul *Wemos* ke *Web hosting*

No	Name	Value
1	valve1	0
2	valve2	1
3	tank1	5
4	total_barang	0
5	pipe1	0
6	pipe2	0
7	pipe3	0
8	pipe4	0
9	pipe5	0
10	pipe6	0
11	elbow1	0
12	elbow2	0
13	elbow3	0
14	elbow4	0
15	bottle_position	137

Gambar 4.15 Tabel Database Pada Web hosting

Status pengiriman dikatakan sukses jika nilai data yang dikirim dari modul *Wemos* seperti pada Gambar 4.14 dapat diterima oleh *Web hosting* yang ditandai dengan berubahnya nilai kolom *Value* pada *Web hosting* tersebut seperti ditunjukkan pada Gambar 4.15.

4.10 Pengujian *Web hosting*

Pada Poin ini yang bertanggung jawab adalah Miftahul Falahi Alfa. Berikut adalah pengujian *web hosting*. Pengujian pada *Web hosting* dilakukan dengan cara melihat status penerimaan data pada *Web hosting* tersebut. Dan dihitung tundaan (*delay*) dari proses pengolahan data pada *Web hosting* tersebut. Hasil pengujian dapat dilihat pada Tabel 4.14

Pengiriman ke *Web hosting* dikatakan sukses jika nilai pada kolom *Value* pada Gambar 4.15 berubah nilainya sesuai dengan data yang dikirim dari modul *Wemos*. Jika nilai pada kolom *Value* pada Gambar 4.15 tidak sesuai dengan data yang dikirim oleh modul *Wemos*, maka proses pengiriman data dikatakan gagal.

Tabel 4.14 Pengujian *Web hosting*

Data	Status Penerimaan Data	<i>Delay</i>
1	Sukses	2
2	Sukses	3
3	Sukses	1
4	Sukses	1
5	Sukses	2

Keterangan : Menggunakan *Server* standar untuk *Web hosting*, dengan alamat "www.waterfilling.dsof-engineering.com" untuk mengakses *Database*

BAB V

PENUTUP

Dari hasil yang telah didapatkan selama proses perancangan dan pembuatan serta proses pengujian dan analisa data untuk Tugas Akhir ini, maka dapat diambil beberapa kesimpulan dan saran yang berguna untuk perbaikan dan pengembangan agar nantinya bermanfaat bagi masyarakat.

5.1 Kesimpulan

1. Pada tugas akhir ini untuk alat pengukur ketinggian cat pada tangki penampungan ditambahkan *sensor level* menggunakan *Sensor Ten Turn Potentiometer*. Dimana *sensor* tersebut mendeteksi ketika tangki kondisi 0% adalah 0,56 Volt dan ketika 100% adalah 4,22 Volt.
2. Pada tugas akhir ini untuk proses pengisian cat kedalam wadah ditambahkan konveyor dan *sensor* pendeteksi wadah untuk membuat sistem menjadi otomatis. Untuk *sensor* pendeteksi wadah menggunakan *Sensor Photodiode*. *Sensor photodiode* lebih sensitif jika permukaan benda untuk pantulan mengkilap dan tidak transparan dan juga tidak terkena sinar yang sangat terang seperti sinar matahari. Tegangan ketika tidak mendeteksi benda sekitar 4,7 sampai 5 Volt dan ketika terdapat benda tegangannya sekitar 2,6 sampai 3,9 Volt.
3. Pada tugas ahir ini untuk dapat memantau proses produksi cat ditambahkan *Software* HMI Android pada ponsel. Untuk penerimaan data di aplikasi Android memiliki *delay* rata-rata 1,25 detik dari kondisi sistem produksi yang bekerja di lapangan. Untuk *software* HMI LabVIEW digunakan pada komputer pusat di ruang kendali. Penerimaan data pada HMI LabVIEW dengan komunikasi *modbus* TCP/IP memiliki *delay* rata-rata 2,15 detik, sedangkan untuk pengiriman data dari *ethernet shield* ke komputer memiliki *delay* rata-rata 3,2 detik. Jadi diperkirakan terjadi *delay* 3,5 detik untuk dapat melihat kondisi sebenarnya sistem produksi yang bekerja dilapangan dengan HMI LabVIEW.

5.2 Saran

Saran untuk penelitian selanjutnya penulis memiliki saran yang diharapkan dapat membuat tugas akhir ini menjadi lebih baik dari apa yang sudah ada.

1. Untuk penelitian selanjutnya pada alat ukur ketinggian cat pada tangki penampungan sebaiknya menggunakan *sensor* selain dari *Ten Turn Potentiometer*, dikarenakan ketika tangki yang digunakan lebih tinggi dari banyaknya putaran *sensor* maka sudah pasti tidak dapat digunakan. Oleh sebab itu sebaiknya mengganti alat ukur tersebut dengan yang lebih universal pada semua tangki, misalkan saja *sensor PING*.
2. Untuk penelitian selanjutnya pada proses pengisian cat kedalam wadah, bagian pendeteksi wadah yang bergerak di konveyor sebaiknya menggunakan selain *sensor photodiode*. Karena *sensor* ini tidak sesuai disemua kondisi, pada tugas akhir ini jika saat cahaya sangat terang maka *sensor* tidak dapat mendeteksi wadah yang bergerak didepannya.
3. pembaharuan untuk aplikasi HMI (*Human Machine Interface*) untuk komputer karena masih lambatnya penerimaan terhadap data baru yang masuk. Dan juga pembaharuan untuk *web hosting* yang sudah menggunakan komputer *server* dengan spesifikasi *hardware* yang tinggi sehingga proses komunikasi dapat berlangsung secara cepat.

DAFTAR PUSTAKA

- [1] Gordon C., Bailey, D. Wright, E., *Practical SCADA Modern Protocol*, IDC Technologies, Australia, 2003
- [2] Schwartz, Marco & Oliver Manickum, *Programming Arduino with LabVIEW*, Packt Publishing, Birmigham, 2015.
- [3] _____, *Industrial Automation / Data Acquisition (DAQ) & Communication / USB Modules / USB-4716* Advantech, Agustus, 2015
- [4] Elektronika Dasar, *Sensor Photodiode*, <http://elektronika-dasar.web.id/sensor-dasar-photodiode/>, 27 januari 2017
- [5] Gita Dwipermata Sari & Abu Hatim Kurniawan, “Alat Monitoring Dan Alarm Denyut Jantung Manusia Terintegrasi Android Berbasis Atmega 328”, *Tugas Akhir*, Program Studi D III Teknik Elektro, Institut Teknologi Sepuluh Nopember, 2013.
- [6] Anwari, M. Dhani M.T. dan Maryanto, Rusmanto, *Panduan Mudah Membuat dan Mengelola Web hosting*, Dian Rakyat, Jakarta, 2003.
- [7] N. S. H, *Android Pemrograman Aplikasi Mobile Smartphone Dan Tablet PC Berbasis Android Revisi Kedua*, Bandung: Informatika Bandung, 2014.
- [8] Sunarfrihantono, Bimo, *PHP dan MySQL untuk Web*, Andi, Yogyakarta, 2002.
- [9] Eko Rudiawan, *Cara Memprogram Wemos D1 R2 Mini ESP8266 Dengan ArduinoIB*, <http://eko-rudiawan.com/cara-memprogram-wemos-esp8266-dengan-arduino/>, 28, januari 2017

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

LAMPIRAN A

A.1 Lampiran Program Arduino

```
#include <SPI.h>
#include <Ethernet.h>
#include "Mudbus.h"
#include <EEPROM.h>

Mudbus Mb;

//===== PHOTO SENSOR & RELAY =====
//int sen1 = A0; //off
int sen2 = A0;
int sen3 = A1;
int sen4 = A2;
int sen5 = A3;
//int sen6 = A4; //off
int sen7 = A5;
int sen_level = A4; //Sensor Level

//===== FOR RELAY MOTOR =====
int relay1 = 2; //Motor Konveyor
int relay2 = 3; //Motor Pendorong
int relay3 = 4; //Motor Konveyor
int relay4 = 5; //Motor Pendorong

//=== FOR VALVE AND PUMP ===
int valve1_bottle = 6; //for bottle filler
int valve2_tank = 9; //for tank filler
int pump1_tank = 7; //for tank filler
int pump2_bottle = 8; //for bottle filler

//For Flag of Photo Sensor
int fb=0, fc=0, fd=0, fe=0, fg=0;
//For WEB HMI
int valve1_status=0;
int valve2_status=0;
int tank_level=50;
int count=0;
int bottle_pos=0;
```

```

char kirim[50];
//For Tank Level
int l;

void setup() {
  //==== ETHERNET MODBUS ====
  //Modbus Communication Setting
  uint8_t mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x51, 0x06 };
  uint8_t ip[] = { 192, 168, 1, 15 };
  uint8_t gateway[] = { 192, 168, 1, 1 };
  uint8_t subnet[] = { 255, 255, 255, 0 };
  Ethernet.begin(mac, ip, gateway, subnet);

  Serial.begin(9600);

  //===== RELAY MOTOR =====
  pinMode(relay1,OUTPUT);
  pinMode(relay2,OUTPUT);
  pinMode(relay3,OUTPUT);
  pinMode(relay4,OUTPUT);
  digitalWrite(relay1,1); //Motor Konveyor Mundur
  digitalWrite(relay2,1); //Motor Pendorong Maju
  digitalWrite(relay3,1); //Motor Konveyor Maju
  digitalWrite(relay4,1); //Motor Pendorong Mundur

  //==== VALVE AND PUMP =====
  pinMode(valve1_bottle,OUTPUT);
  pinMode(valve2_tank,OUTPUT); //
  pinMode(pump1_tank,OUTPUT);
  pinMode(pump2_bottle,OUTPUT);
  digitalWrite(valve1_bottle,1);
  digitalWrite(valve2_tank,1);
  digitalWrite(pump1_tank,1);
  digitalWrite(pump2_bottle,1);

  digitalWrite(relay3,LOW);
}

void loop() {
  Mb.Run();
}

```



```

//==== LEVEL SENSOR ====
int l=analogRead(sen_level);
// int tank_level=map(1,917,988,0,100);
// Serial.print("Level : ");
// Serial.println(l);

//===== CHECK LEVEL TANK =====
if(tank_level >= 99){
    //Pump and Valve are OFF
    digitalWrite(pump1_tank,1);
    delay(1000);
    digitalWrite(valve2_tank,1);
    //Send status of valve1 to HMI Lab View
    //Mb.C[0]=0;
    //Send status to ESP8266
    valve1_status=0;
    sprintf(kirim,"%i:%i:%i:%i:%i:",
valve1_status,valve2_status,tank_level,count,bottle_pos);
    Serial.println(kirim);
    delay(2000);
    //Conveyor forward ON
    digitalWrite(relay3,LOW);
    delay(1000);
    tank_level=50;
}else if(tank_level<=5){
    //Conveyor forward OFF
    digitalWrite(relay3,HIGH);
    //Fill a water to tank
    digitalWrite(pump1_tank,0); //pump1_tank ON until 5 sec
    delay(1000);
    digitalWrite(valve2_tank,0); //Valve1 ON until 5 sec
    //Send status of valve1 to HMI Lab View
    //Mb.C[0]=1;
    //Send status to ESP8266
    valve1_status=1;
    sprintf(kirim,"%i:%i:%i:%i:%i:",
valve1_status,valve2_status,tank_level,count,bottle_pos);
    Serial.println(kirim);
    delay(5000);
}

```

```

    tank_level=99;
}
Mb.R[2]=tank_level;

delay(10);

//==== PHOTO SENSOR GET DATA =====
//int a = analogRead(sen1);
int b = analogRead(sen2);
int c = analogRead(sen3);
int d = analogRead(sen4);
int e = analogRead(sen5);
//int f = analogRead(sen6);
int g = analogRead(sen7);
//Serial.print(a); Serial.print(","); //off
// Serial.print(b); Serial.print(",");
// Serial.print(c); Serial.print(",");
// Serial.print(d); Serial.print(",");
// Serial.print(e); Serial.print(",");
//Serial.print(f); Serial.print(","); //off
// Serial.println(g);

// FOR HMI OF PHOTO SENSOR & BOTTLE POS IN CONVEYOR
//=== Sensor2 detect a goods ===
if(b <= 750 && fb==0){
    //LabView HMI
    for(int i=1;i<=1;i++){
        Mb.R[0]=10;
        //delay(10);
    }
    //Send to ESP8266
    bottle_pos=15;
    sprintf(kirim, "%i:%i:%i:%i:%i:", valve1_status, valve2_status,
tank_level, count,bottle_pos);
    Serial.println(kirim);
    // //Motor Konveyor Maju OFF 1 second
    // digitalWrite(relay3,1);
    // delay(1000);
    // digitalWrite(relay3,0);
    fb=1;
}

```

```

}
//=== Sensor3 detect a goods ===
if(c <= 800 && fb==1 && fc==0){
  //LabView HMI
  for(int i=1;i<=1;i++){
    Mb.R[0]=25;
    //delay(10);
  }
  //Send to ESP8266
  bottle_pos=41;
  sprintf(kirim, "%i:%i:%i:%i:%i:", valve1_status, valve2_status,
tank_level, count,bottle_pos);
  Serial.println(kirim);
  // //Motor Konveyor Maju OFF 1 second
  // digitalWrite(relay3,1);
  // delay(1000);
  // digitalWrite(relay3,0);
  fc=1;
}
//=== Sensor4 detect a goods ===
if(d <= 800 && fc==1 && fd==0){
  //LabView HMI
  for(int i=1;i<=1;i++){
    Mb.R[0]=35;
    //delay(10);
  }
  //Send to ESP8266
  bottle_pos=103;
  sprintf(kirim, "%i:%i:%i:%i:%i:", valve1_status, valve2_status,
tank_level, count,bottle_pos);
  Serial.println(kirim);
  fd=1;
}
//=== Sensor5 detect a goods ===
if(e <= 700 && fd==1 && fe==0){
  //LabView HMI
  for(int i=1;i<=1;i++){
    Mb.R[0]=50;
    //delay(10);
  }
}

```

```

//Send to ESP8266
bottle_pos=137;
sprintf(kirim,"%i:%i:%i:%i:%i",
valve1_status,valve2_status,tank_level,count,bottle_pos);
Serial.println(kirim);
//Conveyor Forward OFF
digitalWrite(relay3,HIGH);
delay(1000);
//Led Full Indicator LabView HMI
for(int i=1;i<=1;i++){
//Mb.C[2]=1;
//delay(10);
}
//Fill a water to bottle
digitalWrite(pump2_bottle,0);
delay(3000);
digitalWrite(valve1_bottle,0);
delay(1000);
//Send status of valve2 to HMI Lab View
//Mb.C[1]=1;
//Send status to ESP8266
valve2_status=1;
sprintf(kirim,"%i:%i:%i:%i:%i",
valve1_status,valve2_status,tank_level,count,bottle_pos);
Serial.println(kirim);
delay(10000);
digitalWrite(pump2_bottle,1);
delay(3000);
digitalWrite(valve1_bottle,1);
delay(3000);
//Send status of valve2 to HMI Lab View
//Mb.C[1]=0;
//Send status to ESP8266
valve2_status=0;
sprintf(kirim,"%i:%i:%i:%i:%i",
valve1_status,valve2_status,tank_level,count,bottle_pos);
Serial.println(kirim);
delay(2000);
//Led Full Indicator LabView HMI
for(int i=1;i<=1;i++){

```

```

    //Mb.C[2]=0;
    //delay(10);
}
//Conveyor Forward ON
digitalWrite(relay3,LOW);
fe=1;
}
//=== Sensor7 derect a goods ===
if(g <= 800 && fe==1 && fg==0){
    count=count+1;
    //LabView HMI
    for(int i=1;i<=1;i++){
        Mb.R[0]=100;
        //delay(10);
    }
    delay(50);
    //Send Data to LabView HMI
    Mb.R[1]=count;
    //Send to ESP8266
    bottle_pos=250;
    sprintf(kirim,"%i:%i:%i:%i:%i",
valve1_status,valve2_status,tank_level,count,bottle_pos);
    Serial.println(kirim);
    //delay(2600);
    //Conveyor Forward OFF
    digitalWrite(relay3,HIGH);
    delay(1000);
    //Driver Motor Forward ON
    digitalWrite(relay2,LOW);
    delay(5100);
    //Driver Motor Forward OFF
    digitalWrite(relay2,HIGH);
    delay(300);
    //Driver Motor Reverse ON
    digitalWrite(relay4,LOW);
    delay(4500);
    //Driver Motor Reverse OFF
    digitalWrite(relay4,HIGH);
    //Conveyor Forward ON
    digitalWrite(relay3,LOW);

```

```
    fg=1;
  }
  //Reset all flags of photo sensor
  if(fb==1 && fc==1 && fe==1 && fd==1 && fg==1){
    fb=0; fc=0; fd=0; fe=0; fg=0;
  }
  delay(50);
}
```

A.2 Lampiran Program Wemos D1 Mini

```
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
SoftwareSerial esp_(D2,D3); //RX, TX
const char* ssid = "RudySetiawan";
const char* password = "rudysetiawan";
byte server[] = {192,168,1,30};
const int httpPort = 80;
const char* host = "www.waterfilling.ds-of-engineering.com";
int f=0,waktu,jumlah;
//===== For Web Parameter =====
int valve1=0, valve2=0, tank_level=33;
int total_barang=0, bottle_pos=0;
int valve1_buf=0, valve2_buf=0,
tank_level_buf=0, total_barang_buf=0, bottle_pos_buf=0;
String data;
void setup() {
  pinMode(D4, OUTPUT);
  digitalWrite(D4, 1);
  Serial.begin(9600);
  esp_.begin(9600);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
    {delay(500);Serial.print(".");}
  Serial.println(WiFi.localIP());
  digitalWrite(D4, 0);
}

void loop(){
  //while (Serial.available()) {
  if(esp_.available()){
    data = esp_.readString();

    //Serial.print(data);
    f=1;
  }
  valve1=getValue(data,':',0).toInt();
  valve2=getValue(data,':',1).toInt();
```

```

    tank_level=getValue(data,',',2).toInt();
    total_barang=getValue(data,',',3).toInt();
    bottle_pos=getValue(data,',',4).toInt();
    if(bottle_pos_buf == 250){ bottle_pos_buf=0; }
    if(total_barang >= total_barang_buf && bottle_pos >=
bottle_pos_buf){
        total_barang_buf=total_barang;
        bottle_pos_buf=bottle_pos;
        // String data2=String(data);
        // int first = data2.indexOf(';');
        // int second= data2.indexOf(':',first+1);
        // String waktu = data2.substring(0,first);
        // String jumlah = data2.substring(first+1,second);
        // Serial.print(valve1);
        // Serial.print(":");
        // Serial.println(valve2);
        // Serial.print(":");
        // Serial.print(tank_level);
        // Serial.print(":");
        // Serial.print(total_barang);
        // Serial.print(":");
        // Serial.print(bottle_pos);
        WiFiClient client;
        // if (client.connect(server, 80)) {
        //     client.print("GET /c/project/konveyor2/add?");
        //     client.print("waktu=");
        //     client.print(waktu);
        //     client.print("&jumlah=");
        //     client.println(jumlah);
        //     client.stop();
        // }
        if (client.connect(host, 80)) {
            //Serial.println("connected");
            // String url = "/konveyor/add?";
            String url = "/update_data.php?";
            url += "valve1=";
            url += valve1;
            url += "&valve2=";
            url += valve2;
            url += "&tank_level=";

```



```

url += tank_level;
url += "&total_barang=";
url += total_barang;
url += "&bottle_pos=";
url += bottle_pos;

// url += "waktu=";
// url += waktu;
// url += "&jumlah=";
// url += jumlah;
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
              "Host: " + host + "\r\n" +
              "Connection: close\r\n\r\n");
//Serial.println("oke");
}
f=0;
}

//Serial.read();
//String second = Serial.readStringUntil(':');
//Serial.print(first);
//Serial.print(">>>>");

//

//}
delay(200);
}

String getValue(String data, char separator, int index){
int found = 0;
int strIndex[] = { 0, -1 };
int maxIndex = data.length()-1;
for(int i=0; i<=maxIndex && found<=index; i++){
if(data.charAt(i)==separator || i==maxIndex){
found++;
strIndex[0] = strIndex[1]+1;
strIndex[1] = (i == maxIndex) ? i+1 : i;
}
}
}

```

```
return found > index ? data.substring(strIndex[0], strIndex[1]) : "";  
}
```

A.3 Lampiran Program Android

```
package com.example.user.waterfilling;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.util.TypedValue;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

public class Hmi extends AppCompatActivity {

    ImageView pipe1, pipe2, pipe3, pipe4, pipe5, pipe6, valve1,
    valve2, elbow1,elbow2,elbow3,elbow4, bottle;
    ProgressDialog progressDialog;
    AlertDialog alertDialog;
    TextView level, tot_barang;
    String
    valve1_val,valve2_val,tank1_val,total_barang_val,pipe1_val,pipe2_val,
    pipe3_val,pipe4_val,pipe5_val,elbow1_val;
    String
    elbow2_val,elbow3_val,elbow4_val,elbow5_val,bottle_pos;
```

```

//For auto Updater
boolean running=false,f=true;
//For Auto Updater
private final Handler mHandler = new Handler();

// CONNECTION_TIMEOUT and READ_TIMEOUT are in
milliseconds
public static final int CONNECTION_TIMEOUT = 10000;
public static final int READ_TIMEOUT = 15000;

// //private ServerSocket serverSocket;
// Handler UIHandler;
// Thread Thread1 = null;
//
// public static final int SERVERPORT = 9000; //Replace with
your choice of PORT number
// public static final String SERVERIP = "192.168.4.1"; //Replace
with your device IP address

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_hmi);

    //Make a Progress Dialog
    progressDialog=new ProgressDialog(this);
    progressDialog.setMessage("Checking Username ...");

    progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
    progressDialog.setIndeterminate(true);
    progressDialog.setCancelable(false);

    //Make an Alert Dialog Builder
    AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(this);
    alertDialogBuilder.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface arg0, int arg1) {
            // Intent i=new Intent(Hmi.this, MainActivity.class);

```

```

//      startActivity(i);
//      finish();
    }
});
AlertDialog=AlertDialogBuilder.create();
AlertDialog.setCancelable(false);

//      UIHandler = new Handler();
//
//      this.Thread1 = new Thread(new Thread1());
//      this.Thread1.start();

//ImageView
pipe1=(ImageView)findViewById(R.id.pipe1);
pipe2=(ImageView)findViewById(R.id.pipe2);
pipe3=(ImageView)findViewById(R.id.pipe3);
pipe4=(ImageView)findViewById(R.id.pipe4);
pipe5=(ImageView)findViewById(R.id.pipe5);
pipe6=(ImageView)findViewById(R.id.pipe6);
elbow1=(ImageView)findViewById(R.id.elbow1);
elbow2=(ImageView)findViewById(R.id.elbow2);
elbow3=(ImageView)findViewById(R.id.elbow3);
elbow4=(ImageView)findViewById(R.id.elbow4);
valve1=(ImageView)findViewById(R.id.valve1);
valve2=(ImageView)findViewById(R.id.valve2);
bottle=(ImageView)findViewById(R.id.bottle);
//TextView
level=(TextView)findViewById(R.id.txt_level);
tot_barang=(TextView)findViewById(R.id.txt_tot_barang);

//new AsyncRetrieve().execute();
startUpdater();
}

//=== Auto Updater Json File ====
private final Runnable mUpdateClock = new Runnable() {
    public void run() {
        // Write Code here
        if(running) {
            new AsyncRetrieve().execute();
        }
    }
}

```

```

        mHandler.postDelayed(mUpdateClock, 500); // 0.5
second
    }
    }
};
public void startUpdater() {
    running=true;
    mHandler.post(mUpdateClock);
}
public void stopUpdater(){
    running=false;
}

private class AsyncRetrieve extends AsyncTask<String, String,
String> {
    ProgressDialog pdLoading = new ProgressDialog(Hmi.this);
    HttpURLConnection conn;
    URL url = null;

    //this method will interact with UI, here display loading
message
    @Override
    protected void onPreExecute() {
        super.onPreExecute();

        pdLoading.setMessage("\tLoading...");
        pdLoading.setCancelable(false);
        //pdLoading.show();

    }

    // This method does not interact with UI, You need to pass
result to onPostExecute to display
    @Override
    protected String doInBackground(String... params) {
        try {
            // Enter URL address where your php file resides
            url = new URL("http://waterfilling.ds-of-
engineering.com/view_data.php");

```

```

    } catch (MalformedURLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return e.toString();
    }
    try {

        // Setup HttpURLConnection class to send and receive
data from php
        conn = (HttpURLConnection) url.openConnection();
        conn.setReadTimeout(READ_TIMEOUT);
        conn.setConnectTimeout(CONNECTION_TIMEOUT);
        conn.setRequestMethod("GET");

        // setDoOutput to true as we receive data from js on file
        conn.setDoOutput(true);

    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
        return e1.toString();
    }
    try {

        int response_code = conn.getResponseCode();

        // Check if successful connection made
        if (response_code == HttpURLConnection.HTTP_OK) {

            // Read data sent from server
            InputStream input = conn.getInputStream();
            BufferedReader reader = new BufferedReader(new
InputStreamReader(input));
            StringBuilder result = new StringBuilder();
            String line;

            while ((line = reader.readLine()) != null) {
                result.append(line);
            }

```

```

        // Pass data to onPostExecute method
        return (result.toString());

    } else {

        return ("unsuccessful");
    }

    } catch (IOException e) {
        e.printStackTrace();
        return e.toString();
    } finally {
        conn.disconnect();
    }

}

// this method will interact with UI, display result sent from
doInBackground method
@Override
protected void onPostExecute(String result) {

    //pdLoading.dismiss();
    if (result.equals("Success! This message is from PHP")) {
        // Toast.makeText(Hmi.this, result.toString(),
        Toast.LENGTH_LONG).show());
    } else {
        // you to understand error returned from doInBackground
method
        //get String (cut)
        try {
            // alertDialog.dismiss();
            val1_val = result.substring(result.indexOf("$")+2,
result.indexOf("#0"));
            val2_val = result.substring(result.indexOf("$")+2,
result.indexOf("#1"));
            tank1_val = result.substring(result.indexOf("$")+2,
result.indexOf("#2"));

```



```

        total_barang_val =
result.substring(result.indexOf("$3")+2, result.indexOf("#3"));
        pipe1_val = result.substring(result.indexOf("$4")+2,
result.indexOf("#4"));
        pipe2_val = result.substring(result.indexOf("$5")+2,
result.indexOf("#5"));
        pipe3_val = result.substring(result.indexOf("$6")+2,
result.indexOf("#6"));
        pipe4_val = result.substring(result.indexOf("$7")+2,
result.indexOf("#7"));
        pipe5_val = result.substring(result.indexOf("$8")+2,
result.indexOf("#8"));
        elbow1_val = result.substring(result.indexOf("$9")+2,
result.indexOf("#9"));
        elbow2_val =
result.substring(result.indexOf("$10")+3, result.indexOf("#10"));
        elbow3_val =
result.substring(result.indexOf("$11")+3, result.indexOf("#11"));
        elbow4_val =
result.substring(result.indexOf("$12")+3, result.indexOf("#12"));
        elbow5_val =
result.substring(result.indexOf("$13")+3, result.indexOf("#13"));
        bottle_pos = result.substring(result.indexOf("$14")+3,
result.indexOf("#14"));

        //Get Value of Tank Level
        level.setText("Level Tank : " + tank1_val + " %");
        tot_barang.setText("Total      Barang      :      "+
total_barang_val);

        if(valve1_val.equals("1")){
            valve1.setVisibility(View.VISIBLE);
            pipe1.setVisibility(View.VISIBLE);
            pipe2.setVisibility(View.VISIBLE);
            pipe3.setVisibility(View.VISIBLE);
            pipe4.setVisibility(View.VISIBLE);
            elbow1.setVisibility(View.VISIBLE);
            elbow2.setVisibility(View.VISIBLE);
        }else{
            valve1.setVisibility(View.INVISIBLE);

```

```

        pipe1.setVisibility(View.INVISIBLE);
        pipe2.setVisibility(View.INVISIBLE);
        pipe3.setVisibility(View.INVISIBLE);
        pipe4.setVisibility(View.INVISIBLE);
        elbow1.setVisibility(View.INVISIBLE);
        elbow2.setVisibility(View.INVISIBLE);
    }
    if(valve2_val.equals("1")){
        valve2.setVisibility(View.VISIBLE);
        pipe5.setVisibility(View.VISIBLE);
        pipe6.setVisibility(View.VISIBLE);
        elbow3.setVisibility(View.VISIBLE);
        elbow4.setVisibility(View.VISIBLE);
    }else{
        valve2.setVisibility(View.INVISIBLE);
        pipe5.setVisibility(View.INVISIBLE);
        pipe6.setVisibility(View.INVISIBLE);
        elbow3.setVisibility(View.INVISIBLE);
        elbow4.setVisibility(View.INVISIBLE);
    }
    if(tank1_val.equals("1")){

    }else{

    }

    if(total_barang_val.equals("1")){

    }else{

    }

    }

    //Movement Control of Bottle
    //Margin Left
    //Start=10, Fill=137, Finish=250
    setMargins(bottle,Integer.parseInt(bottle_pos),0,0,-
55);

    f=true;
}catch (Exception exception){
    if(f){

```

```

        //Toast.makeText(Hmi.this, "Connection Failed!",
Toast.LENGTH_LONG).show();
        alertDialog.setMessage("Connection Failed!");
        alertDialog.show();
        f=false;
    }
}
}

}

}

// Set Margin For Bottle
private void setMargins (View view, int left, int top, int right, int
bottom) {
    if (view.getLayoutParams() instanceof
ViewGroup.MarginLayoutParams) {
        ViewGroup.MarginLayoutParams p =
(ViewGroup.MarginLayoutParams) view.getLayoutParams();
        //Convert PX to DP
        left=(int) TypedValue.applyDimension(
TypedValue.COMPLEX_UNIT_DIP, left,
getResources().getDisplayMetrics());
        top=(int) TypedValue.applyDimension(
TypedValue.COMPLEX_UNIT_DIP, top,
getResources().getDisplayMetrics());
        right=(int) TypedValue.applyDimension(
TypedValue.COMPLEX_UNIT_DIP, right,
getResources().getDisplayMetrics());
        bottom=(int) TypedValue.applyDimension(
TypedValue.COMPLEX_UNIT_DIP, bottom,
getResources().getDisplayMetrics());
        p.setMargins(left, top, right, bottom);
        view.requestLayout();
    }
}
}
}

```

```

// class Thread1 implements Runnable {
//
//     public void run() {
//         Socket socket = null;
//
//         try {
//
//             InetAddress serverAddr =
InetAddress.getByName(SERVERIP);
//             socket = new Socket(serverAddr, SERVERPORT);
//
//             Thread2 commThread = new Thread2(socket);
//             new Thread(commThread).start();
//             return;
//         } catch (IOException e) {
//             e.printStackTrace();
//         }
//     }
// }
//
// class Thread2 implements Runnable {
//
//     private Socket clientSocket;
//
//     private BufferedReader input;
//
//     public Thread2(Socket clientSocket) {
//
//         this.clientSocket = clientSocket;
//
//         try {
//
//             this.input = new BufferedReader(new
InputStreamReader(this.clientSocket.getInputStream()));
//
//         } catch (IOException e) {
//             e.printStackTrace();
//         }
//     }

```

```

//    }
//
//                                public    void    run()    {while
(!Thread.currentThread().isInterrupted()) {
//
//    try {
//
//        String read = input.readLine();
//        if(read != null) {
//            UIHandler.post(new UThread(read));
//        }
//        else {
//            Thread1 = new Thread(new Thread1());
//            Thread1.start();
//            return;
//        }
//    } catch (IOException e) {
//        e.printStackTrace();
//    }
//    }
//    }
//
//
// class UThread implements Runnable {
//     private String msg;
//
//     public UThread(String str) {
//         this.msg = str;
//     }
//
//     @Override
//     public void run() {
//         char control[] = msg.toCharArray();
//
//         if(((int)control[0]) == 1){
//             pipe1.setVisibility(View.VISIBLE);
//         }else if(((int)control[0]) == 0){
//             pipe1.setVisibility(View.INVISIBLE);
//         }
//     }

```

```

//     }
// }

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.back_menu, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle item selection
    switch (item.getItemId()) {
        case R.id.menu_back:
            stopUpdater();
            alertDialog.dismiss();
            finish();
            return true;
        case R.id.menu_refresh:

            default:
            return super.onOptionsItemSelected(item);
    }
}

// public void txt(){
//     Toast.makeText(this,msg,Toast.LENGTH_SHORT).show();
// }
}

```

A.4 Lampiran Program Update Data Web hosting

```
<?php
require "init.php";

$valve1_val = $_GET['valve1'];
$valve2_val = $_GET['valve2'];
$tank_level_val = $_GET['tank_level'];
$total_barang_val = $_GET['total_barang'];
$bottle_pos_val = $_GET['bottle_pos'];

$sql_query1 = "UPDATE
dsorefengi_FINAL_PROJECT.waterFilling SET Value=$valve1_val
WHERE Name='valve1'";
$sql_query2 = "UPDATE
dsorefengi_FINAL_PROJECT.waterFilling SET Value=$valve2_val
WHERE Name='valve2'";
$sql_query3 = "UPDATE
dsorefengi_FINAL_PROJECT.waterFilling SET Value=$tank_level_val
WHERE Name='tank1'";
$sql_query4 = "UPDATE
dsorefengi_FINAL_PROJECT.waterFilling SET
Value=$total_barang_val WHERE Name='total_barang'";
$sql_query5 = "UPDATE
dsorefengi_FINAL_PROJECT.waterFilling SET Value=$bottle_pos_val
WHERE Name='bottle_position'";

$result1 = mysqli_query($con,$sql_query1);
$result2 = mysqli_query($con,$sql_query2);
$result3 = mysqli_query($con,$sql_query3);
$result4 = mysqli_query($con,$sql_query4);
$result5 = mysqli_query($con,$sql_query5);
if (!$result1) { die("Update Data Failed!"); }
if (!$result2) { die("Update Data Failed!"); }
if (!$result3) { die("Update Data Failed!"); }
if (!$result4) { die("Update Data Failed!"); }
if (!$result5) { die("Update Data Failed!"); }

echo "THE DATA HAS BEEN UPDATED.";
mysqli_close($con);
```

```
// Update Data in Database  
// http://waterfilling.ds-of-engineering.com/update\_data.php?valve1=%22a%22&valve2=%22b%22&tank\_level=%22c%22&total\_barang=%22d%22&bottle\_pos=%22e%22  
?>
```


A.5 Lampiran Program *Login* Android

```
<?php
require "init.php";
$name= $_POST["login_name"];
$pass= $_POST["login_pass"];

$sql_query = "select No from
dsofengi_FINAL_PROJECT.employees where Name like '$name' and
Password like '$pass'";
$result = mysqli_query($con,$sql_query);
if(mysqli_num_rows($result) > 0 )
{
    echo "Success";
}else
{
    echo "Login Failed";
}
mysqli_close($con);
?>
```

A.6 Lampiran Program Koneksi ke *Database* di *Web hosting*

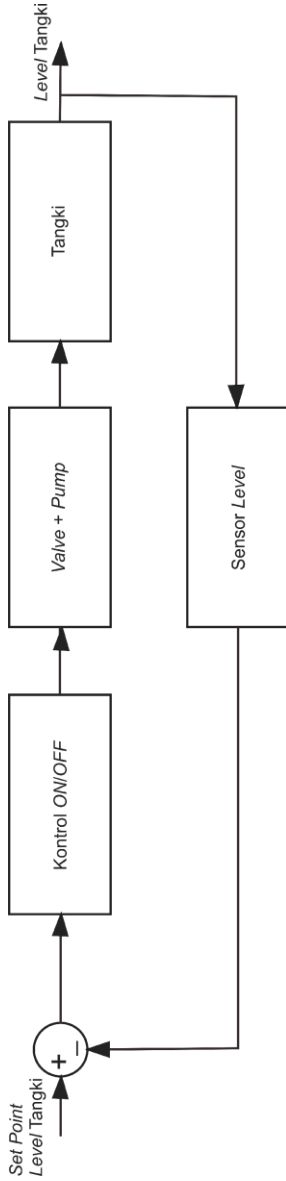
```
<?php
    $host="waterFilling.ds of-engineering.com";
    $user="ds ofengi";
    $pass="62gJdliS18";
    $db="ds ofengi_FINAL_PROJECT";

    $con=mysqli_connect($host,$user,$pass,$db);
    if(!$con){

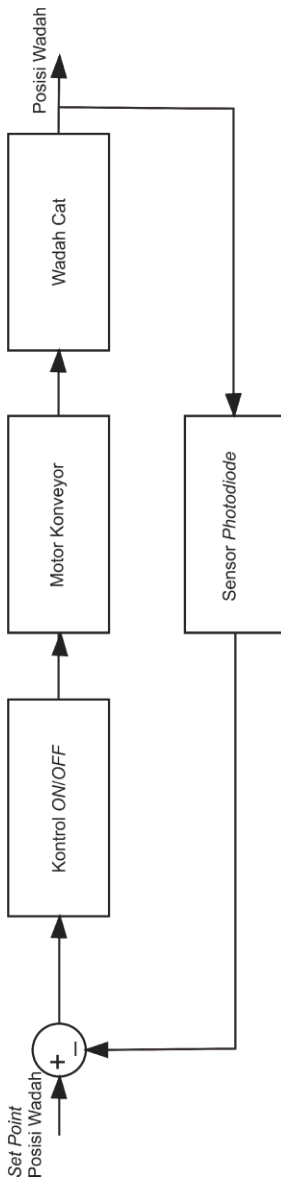
        die("Error in connection .". mysqli_connect_error());

    }
?>
```

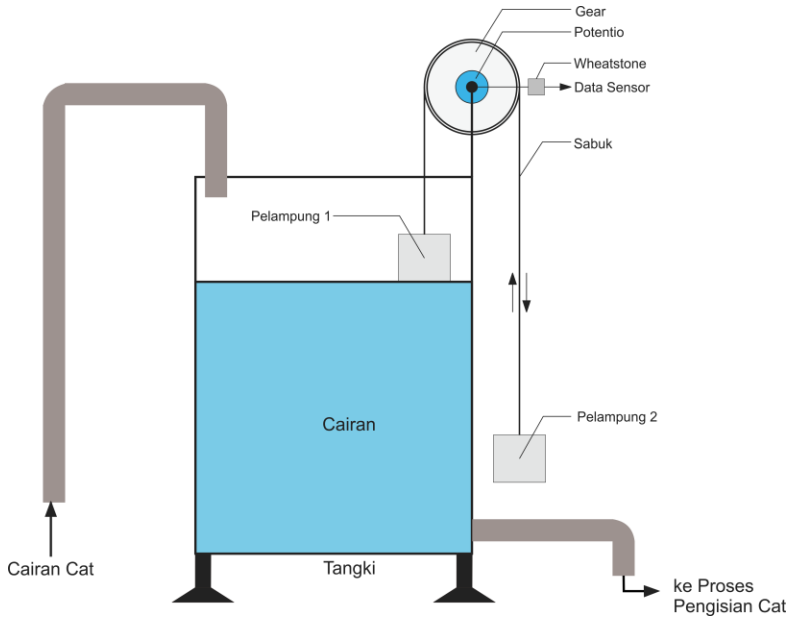
A.7 Lampiran Blok Diagram *Sensor Level*



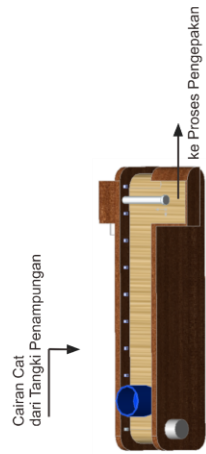
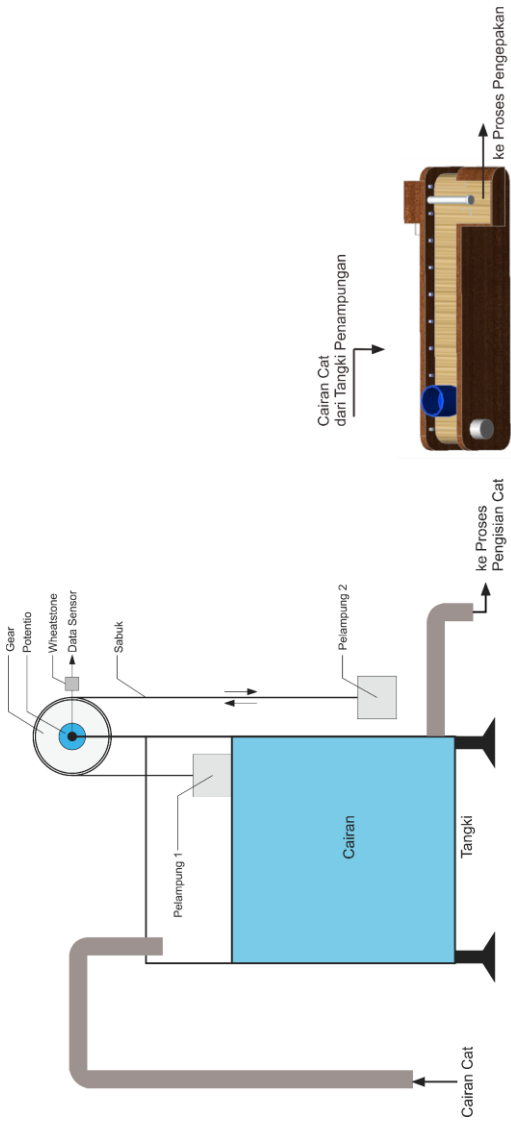
A.8 Lampiran Blok Diagram *Sensor* Pendeteksi Wadah



A.9 Lampiran Blok Fungsional *Sensor Level*



A.10 Lampiran Blok Diagram *Sensor* Pendeteksi Wadah

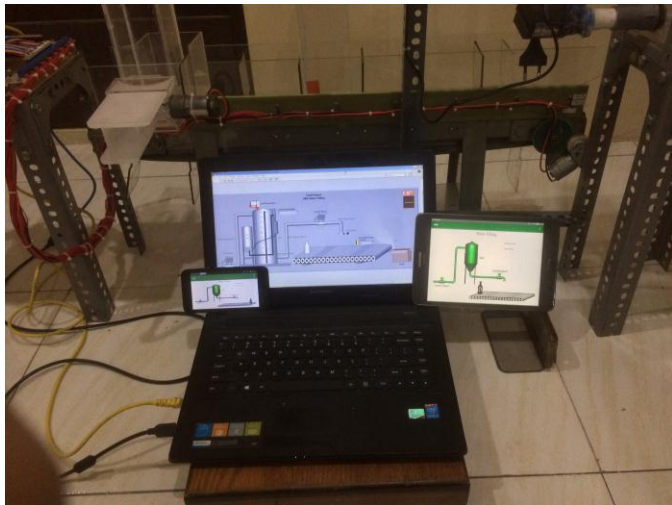


LAMPIRAN B

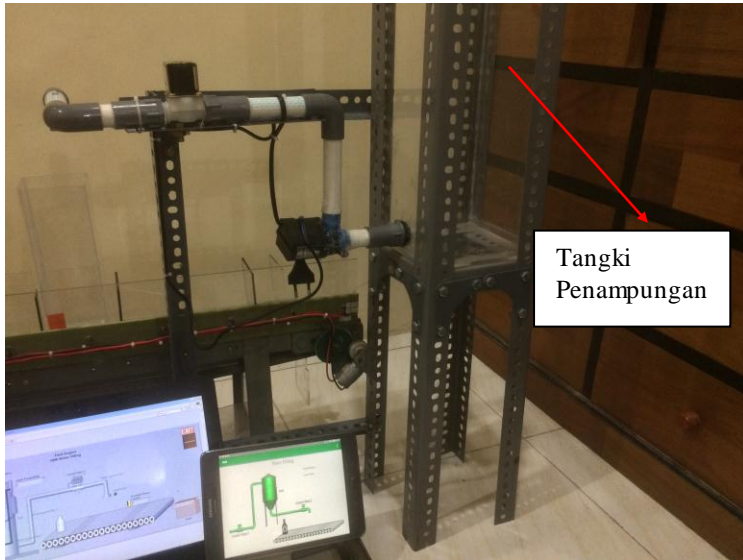
B.1 Dokumentasi Keseluruhan Alat Dari Proses Pengisian Cairan



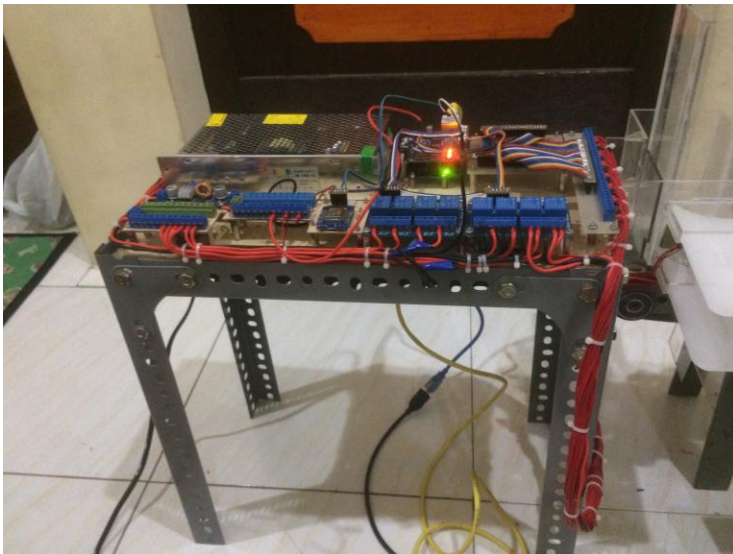
B.2 Dokumentasi Tampilan HMI



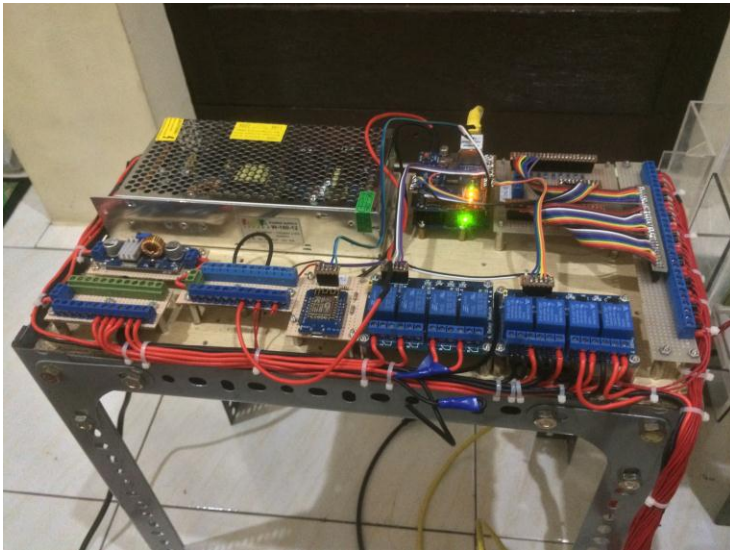
B.3 Dokumentasi Tangki Penampungan



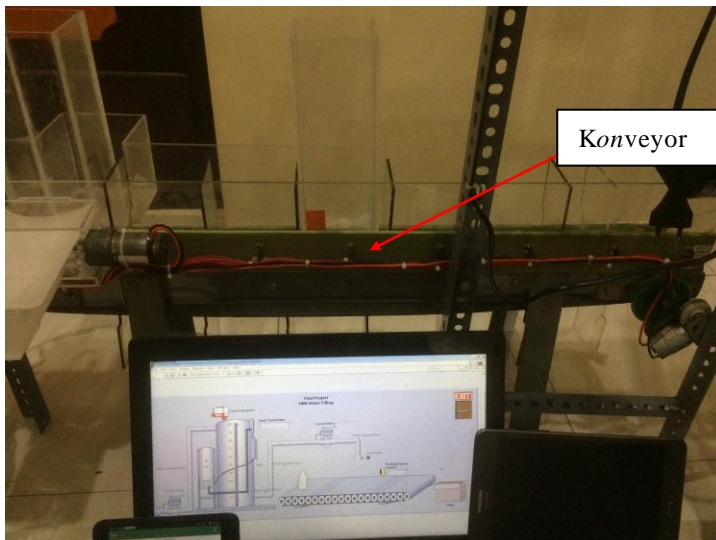
B.4 Dokumentasi Meja Untuk Kontroler Alat Pengisian Cairan



B.5 Dokumentasi Kontroler Untuk Alat Pengisian Cairan



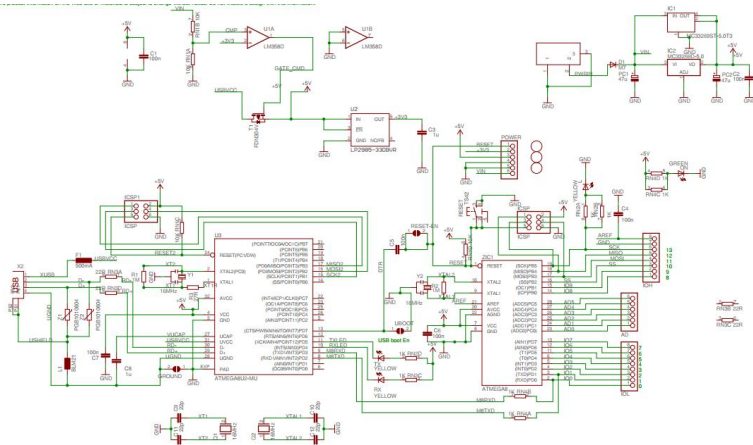
B.6 Dokumentasi Konveyor Untuk Distribusi Wadah



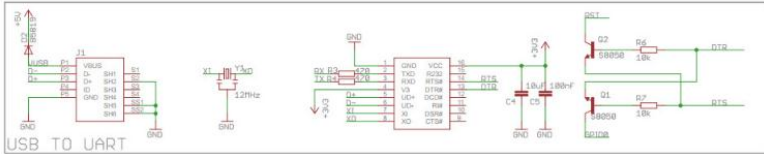
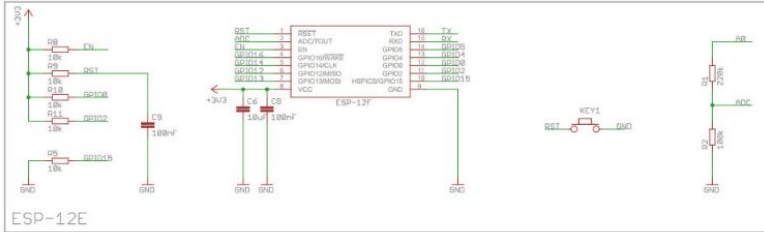
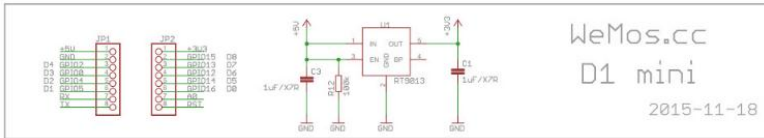
-----Halaman ini sengaja dikosongkan-----

LAMPIRAN C

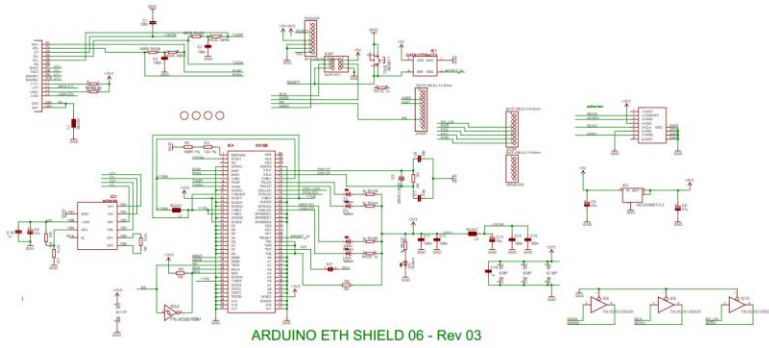
C.1 Datasheet Arduino Uno



C.2 Datasheet Wemos D1 Mini



C.3 Datasheet Ethernet



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

LAMPIRAN D

DAFTAR RIWAYAT HIDUP



Nama : Rudy Setiawan
TTL : Semarang, 20 April 1996
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Jl. Sriwidodo X No 23
Telp/HP : 0822.3489.4829
E-mail : setia.setiawan77@gmail.com

RIWAYAT PENDIDIKAN:

1. 2002-2008 : SD Negeri 04 Purwoyoso Semarang
2. 2008-2011 : SMP Negeri 18 Semarang
3. 2011-2014 : SMK Negeri 4 Semarang
4. 2014-2017 : D3 Teknik Elektro, Program Studi
Komputer Kontrol – Fakultas Vokasi

PENGALAMAN KERJA

1. Kerja Praktek di PT Asia Pacific Fibers Kaliwungu Semarang
2. Kerja Praktek di PT Petro Kimia Gresik

PENGALAMAN ORGANISASI

1. Panitia IARC (*Industrial Automation and Robotic Competition*) kategori PLC (*Programmable Logic Controller*) 2014
2. Panitia IARC (*Industrial Automation and Robotic Competition*) kategori PLC (*Programmable Logic Controller*) 2015
3. Panitia Workshop PLC (*Programmable Logic Controller*) pada acara IARC (*Industrial Automation and Robotic Competition*) 2016

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

DAFTAR RIWAYAT HIDUP



Nama : Miftahul Falahi Alfafa
TTL : Tulungagung, 24 Nov 1995
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Ds. Dono RT 1 RW 1
Sendang Tulungagung
Telp/HP : 0823 3040 3045
E-mail : miftahf77@gmail.com

RIWAYAT PENDIDIKAN:

1. 2002-2008 : MI Miftahul Huda *Dono* Sendang
2. 2008-2011 : MTsN Karangrejo
3. 2011-2014 : SMKN 3 Boyolangu
4. 2014-2017 : D3 Teknik Elektro, Program Studi
Komputer Kontrol – Fakultas Vokasi

PENGALAMAN KERJA

1. Kerja Praktek di PT Roman Ceramic International Mojokerto
2. Kerja Praktek di PT Telkom Akses Sidoarjo

PENGALAMAN ORGANISASI

1. Anggota Himpunan Mahasiswa D3 Teknik Elektro, ITS
2. Anggota Klub Keilmiahan D3 Teknik Elektro, FTI-ITS
3. Panitia IARC (*Industrial Automation and Robotic Competition*)
kategori PLC (*Programmable Logic Controller*) 2016
4. Panitia Workshop PLC (*Programmable Logic Controller*) pada
acara IARC (*Industrial Automation and Robotic Competition*)
2016

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