TUGAS AKHIR – TE 145561

RANCANG BANGUN ALAT DETEKSI KEBOCORAN PIPA DISTRIBUSI AIR BERBASIS SENSOR TEKANAN DAN MIKROKONTROLER

Rifan Syah Rijal
NRP 10311500010038

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M. Eng.
Dwi Lastomo S. Si, M.T.

Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

FINAL PROJECT – TE 145561

PROTOTYPE PIPE LEAKS DETECTOR ON WATER DISTRIBUTION WITH PRESSURE SENSORS AND MICROCONTROLLER

Rifan Syah Rijal
NRP 10311500010038

Supervisor
Ir. Josaphat Pramudijanto, M. Eng.
Dwi Lastomo S. Si, M.T.

Automation Electrical Engineering Department
Faculty of Vocational
Institut Teknologi Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN
TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir kami dengan judul "**RANCANG BANGUN ALAT DETEKSI KEBOCORAN PIPA DISTRIBUSI AIR BERBASIS SENSOR TEKANAN DAN MIKROKONTROLER**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

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

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

Surabaya, 1 Juni 2018

Rifan Syah Rijal
NRP 10311500010038

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

RANCANG BANGUN ALAT DETEKSI KEBOCORAN PIPA DISTRIBUSI AIR BERBASIS SENSOR TEKANAN DAN MIKROKONTROLER

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I	Dosen Pembimbing II
Ir. Josaphat Pramudijanto, M. Eng	Dwi Lastomo,S.Si, M.T.
NIP. 19621005 199003 1 003	NIP. 1987201711059

SURABAYA
JUNI, 2018

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

RANCANG BANGUN ALAT DETEKSI KEBOCORAN PADA PIPA DISTRIBUSI AIR BERBASIS SENSOR TEKANAN DAN MIKROKONTROLER

Nama Mahasiswa : Rifan Syah Rijal
NRP : 10311500010038
Pembimbing I : Ir. Josaphat Pramudijanto, M.Eng.
NIP : 19621005 199003 1 003
Pembimbing II : Dwi Lastomo, S.Si, M.T.
NIP : 1987 2017 11059

ABSTRAK

Distribusi air umumnya dipasang melalui pipa bawah tanah. Memantau jaringan pipa air bawah tanah lebih sulit daripada memantau jalur pipa air yang berada di atas tanah di ruang terbuka. Keadaan ini akan menyebabkan kerugian jika terjadi gangguan pada pipa seperti kebocoran. Kebocoran pada pipa bisa disebabkan oleh beberapa faktor, seperti pipa usia, instalasi yang tidak benar, dan bencana alam. Oleh karena itu, diperlukan solusi untuk mendeteksi dan untuk mengetahui area kerusakan bila terjadi kebocoran. Deteksi area bocor akan menggunakan mekanika fluida berdasarkan data yang diperoleh menggunakan sensor tekanan dan Arduino sebagai mikrokontroler.

Maka dari itu pada Tugas Akhir ini dibuatlah *prototype* alat deteksi kebocoran pipa. Kebocoran dapat diketahui dengan menggunanakan metode pengukuran tekanan air pada sensor tekanan. Metode ini menggunakan tiga sensor tekanan yang dipasang di beberapa titik sebelum dan sesudah kebocoran. Data yang didapat dari pembacaan sensor dikirim ke komputer menggunakan jaringan komunikasi *Ethernet.*

Setelah dilakukan pengujian, hasilnya menunjukkan bahwa tekanan pada pipa menurun setelah terjadi kebocoran. Tekanan yang turun dapat diketahui melalui grafik yang ditampilkan pada *LabVIEW* dengan ketelitian sampai ±10% dari tekanan awal pipa sebelum terjadi kebocoran. Alat simulasi kebocoran ini tidak dapat menahan tekanan diatas ± 20 Psi.

Kata Kunci : Sistem Distribusi Air, Sensor Tekanan, Mikrokontroler.

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

PROTOTYPE PIPE LEAKS DETECTOR ON WATER DISTRIBUTION WITH PRESSURE SENSORS AND MICROCONTROLLER

Student's Name	**: Rifan Syah Rijal**
Regristration Number	**: 10311400010019**
***Supervisor* I**	**: Ir. Josaphat Pramudijanto, M.Eng.**
ID	**: 19621005 199003 1 003**
***Supervisor* II**	**: Dwi Lastomo, S.Si, M.T.**
ID	**: 1987201711059**

ABSTRACT

Water distribution is generally installed through underground pipes. Monitoring underground pipelines is more difficult than monitoring the above-ground water pipelines. This situation will cause losses if there is interruption in the pipeline such as leakage. Leaks in pipes can be caused by several factors, such as age pipes, improper installation, and natural disasters. Therefore, a solution is needed to detect and to know the area of damage in case of leakage. Detection of leaking area will use fluid mechanics based on data obtained using pressure sensors and Arduino as microcontroller.

In this Final Project was made prototype of pipe leak detection tool. Leakage can be determined by using a comparison method of measuring the water pressure. This method uses a 3 pressure sensors mounted at some point before and after the leak. Data obtained from sensor are sent to the computer using an Ethernet communication network and displayed on a PC with LabVIEW software.

After testing, the results show that the pressure on the pipe decreases after a leak. The downward pressure can be determined through the graph shown in LabVIEW with accuracy up to ±10% of the water pressure at the start of the pipe prior to leakage. This leakage simulation tool can not restrain pressure above ± 20 Psi.

Keywords: Water Distribution System, Pressure Sensor, Micrcontroller.

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

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada junjungan kita Nabi Muhammad SAW.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma pada Bidang Studi Elektro Industri, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya.

Dengan terselesaikannya Tugas Akhir ini, Penulis menyampaikan terima kasih yang sebesar - besarnya kepada :

1. Kedua orang tua yang senantiasa mendoakan dan memberikan dukungan dengan tulus tiada henti.
2. Bapak Ir. Josaphat Pramudijanto, M.Eng. dan Bapak Dwi Lastomo, S.Si, M.T. selaku dosen pembimbing 1 dan 2.
3. Pihak Balai Latihan Kerja (BLK) Disnaker Surabaya yang sudah memberi dukungan fasilitas
4. Teman - teman Elektro Industri (Angkatan 2015) yang selalu memberikan doa, semangat, dan dukungannya.
5. Teman - teman Hydra (Angkatan 2015) yang selalu memberikan doa, semangat, dan dukungannya.
6. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.
7. Teman Kontrakan Ceria yang selalu memberikan dukungan

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

Surabaya, Juli 2018

Penulis

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

DAFTAR ISI

HALAMAN JUDUL i
PERNYATAAN KEASLIAN TUGAS AKHIR iii
LEMBAR PENGESAHAN v
ABSTRAK vii
ABSTRACT ix
KATA PENGANTAR xi
DAFTAR ISI xiii
DAFTAR GAMBAR xv
DAFTAR TABEL xvii

BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Perumusan Masalah 2
1.3 Batasan Masalah 2
1.4 Tujuan 2
1.5 Sistematika Laporan 3
1.6 Relevansi 3

BAB II TEORI DASAR PENUNJANG 5
2.1 Teori Dasar Fluida 5
2.2 Persamaan Bernauli 5
2.3 Sensor Tekanan (SKU237545) 6
2.4 Arduino Uno 7
2.5 W-5100 *Ethernet Shield* 9
2.6 LCD 16 x 2 10
2.7 *LabVIEW* 10
2.8 *OPC* 12
2.9 Distribusi Air PDAM 13
2.10 Kebocoran Pipa PDAM 15

BAB III PERANCANGAN DAN PEMBUATAN ALAT 17
3.1 Diagram Fungsional Sistem 17
3.2 Perancangan Perangkat Keras (*Hardware*) 18
3.3 Koneksi *Ethernet Shield* dengan Arduino Uno 21
3.4 Konfigurasi *NI OPC Server* 22
3.4.1 Pengaturan *Channel* 23

3.4.2 Pengaturan *Device* 28
3.4.3 Pengaturan *Tag* 34
3.5 Perancangan Perangkat Lunak (*Software*) 34
3.5.1 Pemrograman Pembacaan Sensor Tekanan 35
3.5.2 Perancangan Software *LabVIEW* 36

BAB IV PENGUJIAN DAN ANALISA ALAT 41
4.1 Kalibrasi Sensor Tekanan 41
4.2 Pengujian Komunikasi *Ethernet* 44
4.3 Pengujian pada *Software LabVIEW* 47
4.4 Pengujian Alat Simulasi 48
4.4.1 Pengujian Tanpa Aliran 48
4.4.2 Pengujian *Stop Valve* 48
4.4.3 Pengujian Kebocoran..... 50
4.5 Pengujian Data *Telemetring LabVIEW*..... 53
4.6 Pengujian Lokasi Kebocoran 58

BAB V PENUTUP 61
5.1 Kesimpulan 61
5.2 Saran 61

DAFTAR PUSTAKA..... 63
LAMPIRAN A PROGRAM ARDUINO A-1
LAMPIRAN B *DATASHEET* B-1
LAMPIRAN C TAMPILAN *LABVIEW* C-1
LAMPIRAN D GAMBAR ALAT D-1
LAMPIRAN E DATA PDAM SURABAYA..... E-1
DAFTAR RIWAYAT HIDUP..... F-1

DAFTAR GAMBAR

Gambar 2.1. Fluida Melewati Pipa yang Tertutup..... 5
Gambar 2.2 Sensor Tekanan..... 6
Gambar 2.3. *Board* Arduino Uno R3..... 9
Gambar 2.4. *Ethernet Shield*..... 9
Gambar 2.5. LCD 16 x 2 10
Gambar 2.6. Tampilan pada *LabVIEW* 11
Gambar 2.7. Diagram Alur *OPC* dengan Perangkat..... 12
Gambar 2.8. Diagram Distribusi Air pada PDAM 14
Gambar 2.9. Kebocoran Pipa PDAM Wisma Mukti 15
Gambar 3.1. Sketsa Diagram Fungsional Alat 17
Gambar 3.2. Perancangan *Prototype* Simulasi Kebocoran Pipa Tampak Samping..... 19
Gambar 3.3. Perancangan *Prototype* Simulasi Kebocoran Pipa Tampak Atas 19
Gambar 3.4. Koneksi Sensor Tekanan ke Arduino Uno..... 20
Gambar 3.5. Koneksi Sensor Tekanan dan LCD..... 21
Gambar 3.6. Koneksi *Ethernet Shileld* pada Arduino Uno 22
Gambar 3.7. *Schematic Ethernet Shield* pada Arduino Uno .. 22
Gambar 3.8. *Software NI OPC Server* 23
Gambar 3.9. *Setting New Channel*..... 23
Gambar 3.10. *Setting* Nama *Channel*..... 24
Gambar 3.11. *Setting* Jenis *Channel*..... 24
Gambar 3.12. *Setting Level Channel* 25
Gambar 3.13. *Setting Network Adapter Channel* 25
Gambar 3.14. *Setting Optimization Channel*..... 26
Gambar 3.15. *Setting Non-Normalized Values Channel*..... 26
Gambar 3.16. *Setting Socket Usage Values Channel* 27
Gambar 3.17 *Setting Summary Channel*..... 27
Gambar 3.18. *Setting New Device* 28
Gambar 3.19. *Setting Name New Device* 28
Gambar 3.20. *Setting Model New Device* 29
Gambar 3.21. *Setting Network Address New Device*..... 29
Gambar 3.22. *Setting Scan Mode New Device*..... 30
Gambar 3.23. *Setting Timing New Device* 30
Gambar 3.24. *Setting Auto Demotion New Device*..... 31
Gambar 3.25. *Setting Database New Device*..... 31

Gambar 3.26. *Setting Ethernet New Device* 32
Gambar 3.27. *Setting Data Acces New Device*..... 32
Gambar 3.28. *Setting Data Encoding New Device*..... 33
Gambar 3.29. *Summary New Device* 33
Gambar 3.30. *Setting Model New Device* 34
Gambar 3.31. *Flowchart* Program Arduino Uno..... 35
Gambar 3.32. *Flowchart* Program *LabVIEW*..... 36
Gambar 3.33. Rancangan *Block Diagram LabVIEW*..... 38
Gambar 3.34. Rancangan *Front Panel LabVIEW* 39
Gambar 4.1. Sambungan *Pressure Gauge* dan Sensor 42
Gambar 4.2. Proses Kalibrasi Sensor Tekanan..... 42
Gambar 4.3. Program pada Arduino Uno ke-1..... 45
Gambar 4.4. *Command Prompt* Pada Laptop dengan IP *address* Arduino Uno ke-1 45
Gambar 4.5. Program pada Arduino Uno ke-2..... 45
Gambar 4.6. *Command Prompt* Pada Laptop dengan IP *address* Arduino Uno ke-2 46
Gambar 4.7. Program pada Arduino Uno ke-3..... 46
Gambar 4.8. *Command Prompt* pada Laptop dengan IP *address* Arduino Uno ke-3 46
Gambar 4.9. Tampilan *LabVIEW* dalam Keadaan *Run*..... 47
Gambar 4.10. Tampilan *LabVIEW* Alat Simulasi Normal..... 55
Gambar 4.11. Tampilan *LabVIEW* Alat Simulasi Kebocoran 1 Dibuka 20^0 56
Gambar 4.12. Tampilan *LabVIEW* Alat Simulasi Kebocoran 1 Dibuka 30^0 56
Gambar 4.13. Tampilan *LabVIEW* Alat Simulasi Kebocoran 2 Dibuka 20^0 57
Gambar 4.14. Tampilan *LabVIEW* Alat Simulasi Kebocoran 2 Dibuka 30^0 58

DAFTAR TABEL

Tabel 2.1 Spesifikasi Sensor Tekanan 7
Tabel 4.1 Hasil Kalibrasi Sensor 1 39
Tabel 4.2 Hasil Kalibrasi Sensor 2 40
Tabel 4.3 Hasil Kalibrasi Sensor 3 40
Tabel 4.4. Data Pengujian Alat Pompa *Off* 45
Tabel 4.5 Tekanan Ketika *Stop Valve* Ditutup $\pm 22^0$ 49
Tabel 4.6 Tekanan Ketika *Stop Valve* Ditutup $\pm 45^0$ 49
Tabel 4.7 Tekanan Ketika *Stop Valve* Ditutup $\pm 54^0$ 49
Tabel 4.8 Tekanan Ketika *Stop Valve* Ditutup $\pm 63^0$ 50
Tabel 4.9 Tekanan Ketika *Stop Valve* Ditutup $\pm 72^0$ 50
Tabel 4.10 Tekanan Ketika *Stop Valve* Ditutup $\pm 54^0$ Tanpa Kebocoran (Normal) 51
Tabel 4.11 Tekanan Ketika *Stop Valve* Ditutup $\pm 54^0$ Kran Kebocoran 1 Dibuka $\pm 20^0$ 51
Tabel 4.12 Tekanan Ketika *Stop Valve* Ditutup $\pm 54^0$ Kran Kebocoran 2 Dibuka $\pm 20^0$ 52
Tabel 4.13 Tekanan Ketika *Stop Valve* Ditutup $\pm 54^0$ Kran Kebocoran 1 Dibuka $\pm 30^0$ 52
Tabel 4.14 Tekanan Ketika *Stop Valve* Ditutup $\pm 54^0$ Kran Kebocoran 2 Dibuka $\pm 30^0$ 53
Tabel 4.15 Tekanan Ketika *Stop Valve* Ditutup $\pm 54^0$ 53
Tabel 4.16 Tekanan Ketika *Stop Valve* Ditutup $\pm 54^0$ Kran Kebocoran 1 Dibuka $\pm 30^0$ 54
Tabel 4.17 Tekanan Ketika *Stop Valve* Ditutup $\pm 54^0$ Kran Kebocoran 2 Dibuka $\pm 30^0$ 55
Tabel 4.18 Pengujian Prediksi Lokasi Kebocoran 1 (Kebocoran 1 Dibuka 30^0) 59
Tabel 4.19 Pengujian Prediksi Lokasi Kebocoran 2 (Kebocoran 2 Dibuka 30^0) 60

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

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pipa adalah benda berbentuk lubang silinder dengan lubang di tengahnya yang terbuat dari logam maupun bahan-bahan lain sebagai sarana pengaliran atau transportasi fluida berbentuk cair, gas maupun udara. Pipa sangat mudah terjadi kebocoran. Kebocoran di pipa mungkin disebabkan oleh beberapa faktor, seperti umur pipa, tidak tepat instalasi, dan kondisi lingkungan (bencana alam, dll).

Menurut Jurnal PDAM, PDAM mengalami kebocoran pipa tersier yang terjadi di Jalan Klampis Anom, Wisma Mukti, Surabaya. Kejadian ini terjadi pada 23 Juli 2018. Karena volume air yang keluar cukup besar maka pipa tersebut terpaksa ditutup dan diperbaiki. Akibatnya beberapa kawasan kehilangan *supply* air bersih selama satu hari dikarenakan perbaikan pipa [1].

Kebocoran sendiri bila dibiarkan lama akan sangat merugikan. Karena semakin lama diketahui adanya bocor maka akan semakin banyak pula fluida yang terbuang. Jika fluida tersebut berupa air maka yang dialami hanya kerugian finansial. Tetapi apabila fluida tersebut adalah gas yang mudah terbakar/meledak maka akan sangat membahayakan. Deteksi kebocoran adalah suata metode untuk mengetahui adanya kebocoran. Alat deteksi kebocoran ini menggunakan mekanika fluida berdasarkan data tekanan yang diperoleh menggunakan sensor tekanan dan Arduino Uno sebagai mikrokontroler. Pipa dalam kondisi bocor memiliki *drop* tekanan dibandingkan tekanan pipa sebelum terjadi kebocoran.

Alat dapat mememonitoring tekanan pada pipa PDAM yang lurus, khususnya pada pipa induk/primer dan pipa sekunder yang memiliki tekanan yang tinggi. Alat ini kurang cocok digunakan pada pipa tersier yang melayani penyadapan langsung ke pelanggan karena tekanan pada pipa tersebut rendah dibawah ±10 Psi.

Sensor Tekanan merupakan salah satu sensor yang digunakan untuk proses monitoring ini. Sensor ini beroperasi dari tekanan 0 – 500 Psi dan menghasilkan tegangan *output linier* antara 0,5 – 4,5 Volt. Sensor tekanan digunakan untuk mendeteksi apakah ada kebocoran pipa saluran melalui perbedaan tekanan pada masing masing jalur pipa. Sehingga diperlukan setidaknya 3 sensor tekanan.

Selain itu dengan adanya 3 sensor tekanan diharapkan alat ini dapat menentukan jalur pipa mana yang mengalami kebocoran sehingga dapat memperkecil area pencarian kebocoran pipa. Selanjutnya data tersebut akan ditampilkan secara *online* melalui komunikasi *Ethernet*.

Alat ini menggunakan mikrokontroler karena mikrokontroler lebih ringkas dibandingkan dengan PLC. Selain itu mikrokontroler lebih hemat biaya. Mikrokontroler adalah sebuah *chip* yang berfungsi sebagai pengontrol rangkaian elektronik dan umunya dapat menyimpan program didalamnya. Mikrokontroler umumnya terdiri dari CPU (*Central Processing Unit*), memori, I/O tertentu dan unit pendukung seperti *Analog-to-Digital Converter* (ADC) yang sudah terintegrasi di dalamnya.

1.2 Perumusan Masalah

Permasalahan yang dihadapi pada Tugas Akhir ini adalah sulitnya mendeteksi kebocoran pipa. Kebocoran pipa bila dibiarkan berlarut-larut dapat menyebabkan kerugian yang besar karena material (fluida) yang terbuang secara terus menerus.

1.3 Batasan Masalah

Batasan masalah Tugas Akhir ini antara lain

- Alat ini adalah prototype/rancangan alat deteksi kebocoran dengan menggunakan sensor tekanan
- Tekanan yang diberikan maksimal adalah sesuai dengan daya tahan pipa PVC (±25 Psi)
- Air pada saluran didapatkan dari pompa air
- Kebocoran hanya terjadi pada salah satu titik saja (tidak ada kebocoran ganda).
- Tekanan air yang diakibatkan ketinggian air dari tanah dalam pipa saluran diabaikan.

1.4 Tujuan

Tujuan yang akan dicapai dari Tugas Akhir ini adalah sebagai berikut:

- Membuat Alat Deteksi Kebocoran Pipa dengan sensor tekanan.
- Merancang Alat yang mampu melakukan Monitoring kebocoran pipa.

- Mendesain Alat yang mampu mendeteksi kebocoran pipa.

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, konsep dari fluida, persamaan kontinuitas, persamaan bernauli sensor tekanan, Arduino Uno, dan *LabVIEW*.

Bab III Perancangan Sistem

Bab ini membahas desain dan perancangan alat mekanik dan elektrik.

Bab IV Simulasi, Implementasi dan Analisa Sistem

Bab ini memuat hasil simulasi dan implementasi serta analisis dari hasil tersebut.

Bab V Penutup

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

1.6 Relevansi

Jika Tugas Akhir ini selesai maka bisa digunakan alat bantu untuk mendeteksi kebocoran atau digunakan sebagai modul praktikum. Selanjutnya *prototype* yang sudah dibuat ini bisa dikembangkan pihak PDAM untuk membuat alat deteksi kebocoran berbasis sensor tekanan. Jika jarak pipa panjang dapat dipasang beberapa sensor untuk mempersempit area pencarian kebocoran pipa. Selain itu alat ini dapat juga digunakan sebagai deteksi kebocoran tidak hanya air, tetapi bisa juga minyak, gas dan material industri.

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

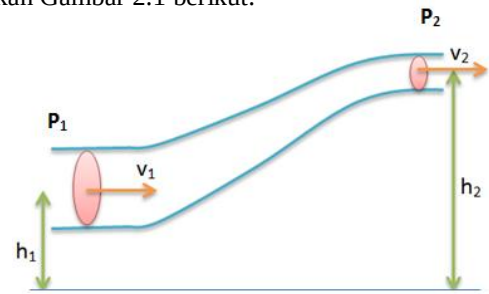
BAB II
TEORI DASAR PENUNJANG

2.1 Teori Dasar Fluida [2]

Fluida adalah suatu zat yang bisa mengalami perubahan-perubahan bentuknya secara *continue*/terus-menerus bila terkena tekanan/gaya geser walaupun relatif kecil atatu bisa juga dikatakan suatu zat yang mengalir. Fluida (*fluid*) adalah zat yang dapat mengalir. Jadi istilah fluida termasuk cairan dan gas. Zat cair gan gas tidak mempertahankan bentuk yang tetap. Keduanya memiliki kemampuan untuk mengalir. Dengan demikian keduanya disebut sebagai fluida. Berdasarkan kondisinya fluida dibagi menjadi 2 keadaan yaitu fluida statis dan fluida dinamis. Fluida statis adalah fluida yang berada kondisi diam dan tidak bergerak. Contohnya air sumur, air dalam gelas dan air laut. Fluida dinamis adalah fluida yang berada dalam kondisi bergerak atau mengalir. Contohnya adalah aliran air dan angin. Dari fluida dinamis di temukan energi potensial yang dapat dijadikan sumber energi listrik.

2.2 Persamaan Bernauli [3]

Prinsip Bernoulli adalah istilah di dalam mekanika fluida yang menyatakan bahwa pada suatu aliran fluida, peningkatan pada kecepatan fluida akan menimbulkan penurunan tekanan pada aliran tersebut. Prinsip ini diambil dari nama ilmuwan Belanda/Swiss yang bernama Daniel Bernoulli. Aliran dari suatu segmen fluida ideal yang melewati pipa tidak beraturan dalam selang waktu Δt ditunjukkan Gambar 2.1 berikut.



Gambar 2.1. Fluida Melewati Pipa yang Tertutup

Persamaan Bernoulii sebagaimana dapat diterapkan pada fluida ideal. Persamaan ini juga dapat ditulis:

$P + ½ \rho v_2 + \rho gh = konstan$ ..... (2.1)

Persamaan ini menunjukkan bahwa tekanan fluida berkurang ketika kelajuan fluida bertambah. Selain itu, tekanan juga berkurang ketika ketinggiannya bertambah.

2.3 Sensor Tekanan (SKU237545) [4]

Sensor tekanan mengukur tekanan, biasanya gas atau cairan. Tekanan adalah ekspresi dari gaya yang dibutuhkan untuk menghentikan cairan dari perluasan, dan biasanya dinyatakan dalam hal gaya per satuan luas. Gambar 2.2 berikut adalah gambar sensor tekanan yang digunakan pada Tugas Akhir ini.

Gambar 2.2. Sensor Tekanan

Sensor tekanan adalah sensor untuk mengukur tekanan suatu zat. Tekanan (p) adalah satuan fisika untuk menyatakan gaya (F) per satuan luas (A). Satuan tekanan sering digunakan untuk mengukur kekuatan dari suatu cairan atau gas. Prinsip kerja dari sensor tekanan ini adalah mengubah tegangan mekanis menjadi sinyal listrik. kurang ketegangan didasarkan pada prinsip bahwa tahanan pengantar berubah dengan panjang dan luas penampang. Spesifikasi sensor tekanan yang digunakan pada Tugas Akhir ini dijelaskan pada Tabel 2.1.

Tabel 2.1 Spesifikasi Sensor Tekanan

Working Voltage	*DC 5,0 Volt*
Output Voltage	*DC 0,5-4,5 Volt*
Sensor material	*Carbon steel alloy*
Working Current	*≤10 mA*
Working Pressure Range	*0-1,2 MPa*
The Biggest Pressure	*2,4 MPa*
Cable length	*19cm*
Destroy Pressure	*3,0 MPa*
Working TEMP. Range	*0-85℃*
Storage Temperature Range	*0-100℃*
Measuring Error	*±1,5 %FSO*
Temperature Range Error:	*±3,5 %FSO*
Response Time	*≤2,0 ms*
Cycle Life	*500,000 pcs*
Application	*non-corrosive gas liquid measurement*

High quality Pressure Transducer Sensor Oil Fuel for Gas Water Features:
Easy removal, carbon steel connection more firmly. Stainless steel is durable, sealed waterproof line, imported chips. Wiring: red +, black -, yellow output

2.4 Arduino Uno [5]

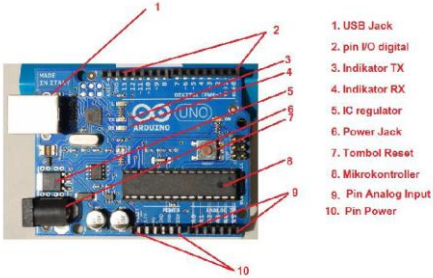
Arduino Uno R3 adalah papan pengembangan mikrokontroler yang berbasis chip ATmega328P. Arduino Uno memiliki 14 *digital* pin *input / output* (atau biasa ditulis I/O, dimana 14 pin diantaranya dapat digunakan sebagai *output* PWM antara lain pin 0 sampai 13), 6 pin *input analog*, menggunakan *crystal* 16 MHz antara lain pin A0 sampai A5, koneksi USB, jack listrik, *header* ICSP dan tombol *reset*.

Arduino terdiri dari dua bagian utama: Arduino papan, yang merupakan bagian dari perangkat keras ketika bekerja pada saat kita membangun objek dan Arduino IDE, bagian perangkat lunak yang dijalankan pada komputer. Dengan menggunakan IDE untuk membuat sketsa (program komputer kecil) kemudian diupload ke papan Arduino. sketsa memberitahu papan apa yang harus dilakukan.

Salah satu jenis Arduino yang menggunakan mikrokontroler tipe Atmel AVR (8-*bit*) adalah Arduino Uno. Arduino Uno adalah papan mikrokontroler berbasiskan ATmega328. Spesifikasinya adalah seperti berikut:

- Memiliki kecepatan *clock* mencapai 16MHz.
- Memiliki *Flash Memory* 32 KB, 0,5 KB diantaranya sebagai *bootloader*.
- Rentang *input* sumber tegangan 7-20 Volt, dan beroperasi pada tegangan 5 Volt.
- Memiliki EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 1 KB.
- Memiliki SRAM (*Static Random Access Memory*) sebesar 2 KB.
- Memiliki pin I/O *digital* sebanyak 14 pin dan 6 pin diantaranya digunakan sebagai *output* PWM (*Pulse Width Modulation*).
- Memiliki pin I/O *analog* sebanyak 6 pin.
- Arus DC per pin I/O 40 mA dan pada pada pin 3,3 Volt 50 mA.

Letak pin pada *board* Arduino Uno dapat dilihat pada Gambar 2.3.



Gambar 2.3. *Board* Arduino Uno R3

2.5 W-5100 *Ethernet Shield* [6]

Ethernet Shield merupakan suatu perangkat yang dapat menambah kemampuan Arduino untuk terhubung ke jaringan computer. *Ethernet shield* berbasiskan *chip Ethernet Wiznet* W5100. *chip Ethernet Wiznet* W5100 ini menyediakan jaringan internet (IP) baik TCP dan UDP. Dan didukung oleh 4 soket koneksi yang simultan. *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 (*Secure Digital*), yang dapat digunakan untuk menyimpan file yang diakses melalui jaringan. *Board* Arduino dapat berkomunikasi dengan *chip* Winzet W5100 dan SD *card* menggunakan bus SPI (*Serial Peripheral Interface*). Bus SPI menggunakan pin *digital* 11, 12, dan 13 pada Arduino Uno. Pin *digital* 10 digunakan untuk memilih W5100 dan pin *digital* 4 digunakan untuk memilih SD *card*. Pin yang sudah disebutkan tidak dapat digunakan untuk *input/output* ketika kita menggunakan *Ethernet Shield*. Karena W5100 dan SD *card* berbagi bus SPI, hanya salah satu yang dapat aktif pada satu waktu.

Gambar 2.4. *Ethernet Shield*

2.6 LCD 16 x 2 [7]

Liquid Crystal Display (*LCD*) pada Gambar 2.5 adalah suatu *display* dari bahan cairan kristal yang pengoperasiannya menggunakan sistem dot matriks. *LCD* banyak digunakan sebagai *display* dari alat-alat elektronik seperti kalkulator, *multitester digital*, jam *digital*, dan sebagainya. *LCD* yang dipakai pada pembuatan alat ini adalah 2 baris x 16.

Gambar 2.5. LCD 16 x 2

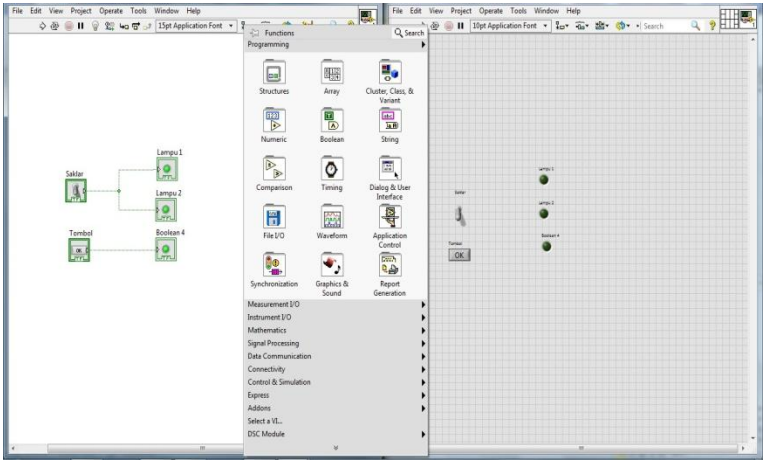
2.7 *LabVIEW* [6]

LabVIEW adalah berorientasi objek grafis bahasa komputer yang dikembangkan untuk memfasilitasi perangkat keras dan perangkat lunak. Komunikasi *LabVIEW* adalah bahasa komputer yang lengkap yang dapat digunakan seperti *Basic*, FORTRAN, atau C. Dalam beberapa tingkatan instrumentasi *virtual* akuisisi data yang membuatnya mudah untuk mengontrol aliran data, dan pemrosesan sinyal banyak dan algoritma analisis datang dengan perangkat lunak sebagai instrumentasi *virtual premade*. Di dalam kelas, kesamaan antara instrumen *virtual* dan nyata membantu mahasiswa memahami bagaimana informasi dilewatkan antara komputer dan instrumen terpasang.

Program pada *LabVIEW* yang disebut sebagai VI (*Virtual Instrument*). Setiap VI menggunakan fungsi-fungsi yang menggerakkan masukan dari pemakai antarmuka atau sumber lain dan menampilkan informasi itu atau memindahkannya ke file lain atau ke komputer lain. VI menyediakan tiga komponen yaitu :

Panel muka (*front panel*), yang melayani antarmuka pengguna. *Front panel* merupakan *interface* antar pengguna (*user*) dengan program. Di dalam *front panel* terdapat kontrol (*input*) dan indikator

(*output*) yang dikenal dengan Kontrol Palet. Kontrol pada *front panel* dapat berupa knop, tombol, *dial*, dan lainnya. Sedangkan untuk indikator (*output)* dapat berupa LED, grafik, dan tampilan lainnya. Kontrol menirukan *input* instrumen dan menyuplai data ke diagram blok pada VI yang bersangkutan. Indikator menirukan instrumen keluaran dan menampilkan data. Untuk menampilkan *control panel* maka pada *front panel* dapat diaktifkan *View >> Control Pallete.*



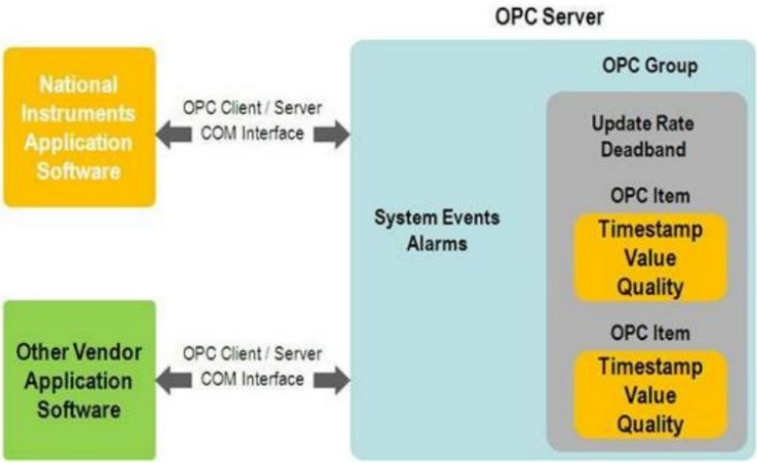
Gambar 2.6. Tampilan pada *LabVIEW*

Diagram blok, berisi kode sumber grafis yang menggambarkan fungsi-fungsi VI. Setelah membangun *front panel*, perlu menambahkan kode menggunakan grafis yang mewakili fungsi untuk mengendalikan objek - objek *front panel*. Blok diagram berisi *source code* grafis ini. Objek – objek *front panel* tampak seperti terminal pada diagram blok. *Virtual Instrument* menunjukkan beberapa objek diagram blok seperti terminal – terminal, fungsi – fungsi dan alur. Pada diagram blok terdapat *Function Pallete*, yang menyediakan *visible items* untuk membentuk suatu program. Jika tidak ditampilkan pada diagram blok *windows* maka dapat diaktifkan melalui *Window >> Show Functions Pallete*, atau klik

kanan area pada diagram *window*. Pada Gambar 2.6 merupakan tampilan *front panel* dan *block diagram* pada *LabVIEW*

2.8 *OPC* [8]

OPC (*OLE for Procces Control*) dan sedangkan kepanjangan *OLE* adalah *Object Linking* and *Embeded*, merupakan bentuk dari tampilan antar muka yang umumnya di gunakan diantara berbagai sumber data yang ada seperti *Programmable Logic Controller* (PLC), *Remote Terminal Unit* (RTU) dan sensor- sensor yang ada di pabrik memlalui HMI / SCADA. Dengan menggunakan *OPC*, *software* yang anda gunakan sebagai HMI dapat melakukan komunikasi data dengan perangkat keras anda (*hardware*) tanpa harus menduplikat *device driver*.



Gambar 2.7. Diagram Alur *OPC* dengan Perangkat

2.8.1 *OPC Server* [8]

Konsep dasar dari *OPC Server* adalah kita memiliki *OPC Server* dan beberapa *OPC client* untuk berkomunikasi dengan *server* tersebut dan untuk menulis data atau membaca data tersebut. Disaat keadaan tingkat tinggi *OPC Server* terbagi beberapa komponen yaitu *server*, grup, dan barang. *OPC Server* tersebut mempunyai informasi tentang *server* dan menjadi wadah untuk grup objek *OPC*.

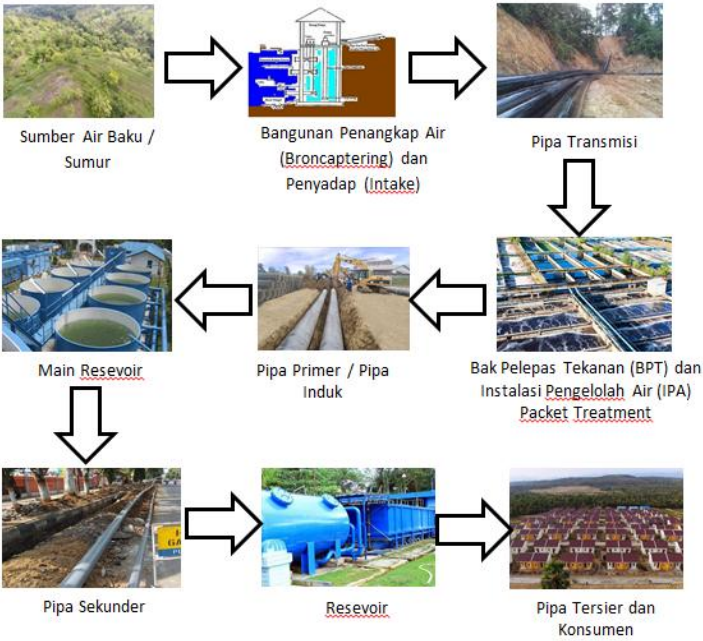
OPC Group menyediakan jalan untuk menghubungkan *clients* untuk mengatur data tersebut. Contohnya, Grup tersebut mampu menampilkan beberapa barang dalam tampilan operator tertentu dan mampu membaca dan menulis data tersebut. Meskipun *OPC* ditujukan untuk mengakses data pada *server* tertentu namun *interfaces OPC* dapat digunakan diberbagai tempat dengan menggunakan aplikasi yang beragam. Di titik terendah, mereka dapat mengambil data yang masih mentah lalu mengirimnya ke SCADA atau DCS. Arsitektur dan desain *OPC* dapat membuat *OPC client* dapat mengakses data dari berbagai *OPC Server* yang menyediakan berbagai jenis vendor *OPC* yang menjalankan berbagai macam jenis cara.

2.8.2 *NI OPC Server* [8]

NI OPC Server merupakan produk dari *National Instrument* untuk menghubungkan dengan perangkat keras sehingga dapat melakukan komunikasi dengan aplikasi anda, namun kebanyakan *NI OPC Server* dihubungkan dengan *LabVIEW*. Sehingga kombinasi dari *LABVIEW* dan *NI OPC Server* dapat menyediakan sebuah wadah yang dapat menampilkan kemampuan tingkat tinggi dalam hal pengukuran dan kontrol di dunia industri dan pabrik. *NI OPC Server* terhubung melalui *OPC Client* yang terdapat di *LabVIEW Data logging* dan *Supervisory Control* (DCS) *Module* yang dapat mengembangkan HMI atau SCADA yang sepenuhnya matang dan dilengkapi oleh PLC, DCA, dan sensor- sensor.

2.9 Distribusi Air PDAM [9]

PDAM atau Perusahaan Daerah Air Minum merupakan salah satu unit usaha milik daerah, yang yang bergerak dalam distribusi air bersih bagi masyarakat umum. PDAM terdapat di setiap provinsi, kabupaten, dan kotamadya di seluruh Indonesia. PDAM merupakan perusahaan daerah sebagai sarana penyedia air bersih yang diawasi dan dimonitor oleh aparat-aparat eksekutif maupun legislatif daerah. PDAM sendiri memiliki alur distribusi air yang secara garis besarnya terdapat pada Gambar 2.8



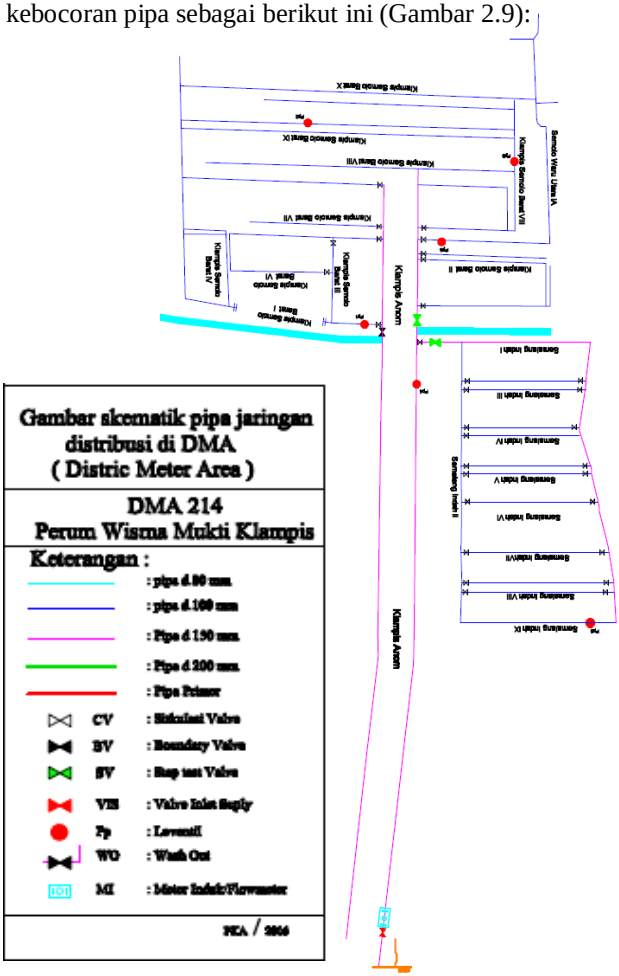
Gambar 2.8. Diagram Distribusi Air pada PDAM

Pada Gambar 2.8, air dari PDAM didapatkan dari Sumber Air / Sumur. Kemudian air tersebut akan disadap oleh bangunan *Intake* yang kemudian disalurkan ke pipa transmisi. Kemudian air dari pipa transmisi akan ditampung pada Bak Pelepas Tekanan (BPT) dan Instalasi Pengelolahan Air (IPA). Lalu disalurkan ke *Main Resevoir* melalui pipa primer/induk. Dari *Main Resevoir* akan didistribusikan ke *Resevoir* lainnya melalui pipa sekunder. Pada *Reservoir* distribusi air akan dilanjutkan ke konsumen air.

Alat ini diharapkan dapat mememonitoring tekanan pada pipa PDAM, khususnya pada pipa transmisi, pipa induk dan pipa sekunder yang memiliki tekanan yang tinggi. Alat ini kurang cocok digunakan pada pipa tersier yang melayani penyadapan langsung ke pelangkan karena tekanan pada pipa tersebut rendah dibawah ±10 Psi.

2.10 Kebocoran Pipa PDAM [1]

Dari jurnal PDAM Kota Surabaya didapatkan kejadian kebocoran pipa sebagai berikut ini (Gambar 2.9):



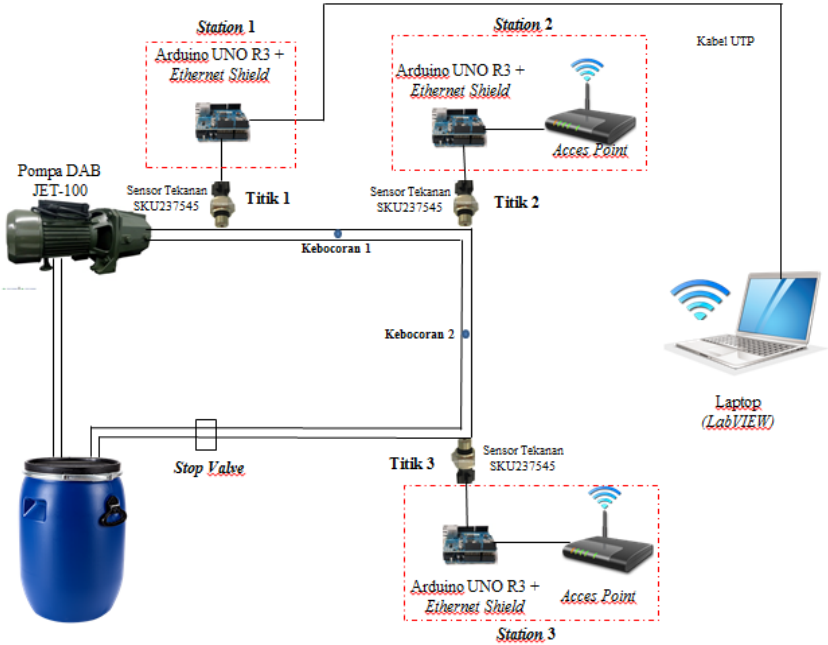
Gambar 2.9. Kebocoran Pipa PDAM Wisma Mukti

Pada diagram tersebut (Gambar 2.9) terjadi kebocoran pipa pada kawasan Wisma Mukti, Surabaya. Kawasan tersebut mendapat air dari Instalasi Pengelolahan Air (IPA) Ngagel. PDAM menggunakan metode DMA (*Distributed Meter Area*) untuk mengetahui dimana letak lokasi kebocoran pipa. DMA sendiri terdiri dari 2 variabel utama yaitu tekanan dan debit air. Titik merah digunakan sebagai titik cek tekanan dan kotak biru digunakan sebagai titik cek debit. Diameter pipa yang digunakan di Wisma Mukti sendiri bervariasi tergantung pada jenisnya apakah pipa tersebut merupakan pipa primer, sekunder atau tersier. Alat ini dapat diaplikasikan untuk pipa sekunder dan primer (tekanan diatas ±10 Psi).

BAB III
PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini berisi tahapan mengenai tahapan yang dilakukan dalam perencanaan dan pembuatan tugas akhir. Penjelasan diawali dengan blok fungsional sistem secara keseluruhan yang meliputi proses kerja alat dalam bentuk alur diagram. Perancangan mekanik membahas mengenai desain dan pembuatan alat simulasi. Perancangan *software* membahas mengenai pembuatan *software* IDE Arduino dan desain *Human Machine Interface* (HMI) menggunakan *software LabVIEW*.

3.1 Diagram Fungsional Sistem



Gambar 3.1. Sketsa Diagram Fungsional Alat

Sebelum melakukan perancangan perancangan perangkat keras dan perangkat lunak, diperlukan sebuah perancangan blok fungsional sistem berupa blok diagram yang menjelaskan kerja secara keseluruhan tugas akhir ini. Secara keseluruhan blok fungsional sistem dapat dilihat pada Gambar 3.1.

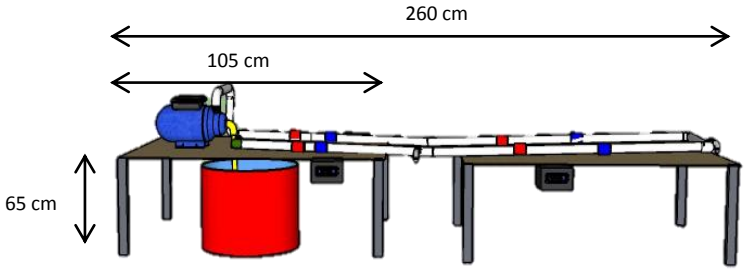
Pada Gambar 3.1 tersebut pompa air digunakan untuk menghasilkan tekanan pada pipa saluran air. Gabungan Arduino Uno, *Ethernet Shield*, dan LCD diumpamakan sebagai *station*. Terdapat 3 titik yang dimonitoring tekanan airnya yaitu titik 1,2 dan 3. Titik 1 dihubungkan ke *station* 1, titik 2 dihubungkan ke *station* 2 dan titik 3 dihubungkan ke *station* 3. Laptop diumpamakan sebagai *server*. *Station* 1 berkomunikasi dengan *server* menggunakan media kabel UTP tanpa melewati *Acces Point* (router). Sedangkan untuk *Station* 2 dan 3 menggunakan *Acces Point* yang kemudian data ditransmisikan secara *wireless* ke *server*. Diantara titik 1 dan 2 terdapat kebocoran 1 yang dapat diputar antara sudut 0^0-90^0. Diantata titik 2 dan 3 terdapat kebocoran 2 yang dapat diputar antara sudut 0^0-90^0. *Stop valve* digunakan untuk memperbesar tekanan dalam pipa dan dapat diputar antara sudut 0^0-90^0. Air yang keluar dari *stop valve* akan masuk tanki air dan nantinya dipompa kembali.

3.2 Perancangan Perangkat Keras (*Hardware*)

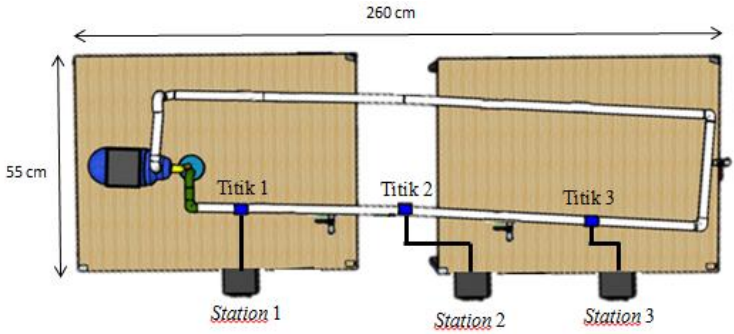
Pada perancangan *hardware* pada tugas akhir ini yang dibahas terdiri dari perancangan *prototype* simulasi kebocoran pipa dan perancangan elektrik berupa rangkaian sensor beserta komunikasinya.

3.2.1 Perancangan *Prototype* Simulasi Kebocoran Pipa

Perancangan *prototype* simulasi kebocoran pipa ini dimaksudkan untuk jarak yang jauh dan dengan kondisi lingkungan yang seperti pada saluran distribusi air.



Gambar 3.2. Perancangan *Prototype* Simulasi Kebocoran Pipa Tampak Samping



Gambar 3.3. Perancangan *Prototype* Simulasi Kebocoran Pipa Tampak Atas

Bahan bahan yang digunakan untuk pembuatan alat simulasi kebocoran di atas antara lain:

1. Pompa Air
2. Tanki Air ukuran sedang
3. Pipa *PVC* ½ dim ±5 m
4. Lem *PVC*
5. Besi Lubang Tipis ±12 m
6. Triplek 6mm ± 6 m^2
7. *Water* Mur ±12 buah

8. *Tee* drat dalam 7 buah
9. *Knee* ±8 buah
10. Kran untuk kebocoran 3 buah
11. Selang 5/8 ±6 m

Dimensi alat simulasi:

1. Panjang meja : 1,05 m
2. Lebar meja : 0,5 m
3. Panjang pipa : 2,1 m
4. Lebar pipa : 0,3 m
5. Panjang Total Pipa : 5,2 m

Rancangan *prototype* di atas menggunakan bak penampungan air dengan diameter 45 cm. Untuk pompa digunakan pompa DAB JET-100 dengan tekanan maksimal adalaah ±50 Psi. Meja dibuat dari bahan triplek tebal 6 mm dengan rangka menggunakan besi. Ukuran pipa yang digunakan sebesar 1/2", sedangkan total panjang pipa yang digunakan sebesar ±520 cm. Sensor tekanan diletakkan pada ±52 cm, ±157cm dan ± 260 cm dari ujung pipa.

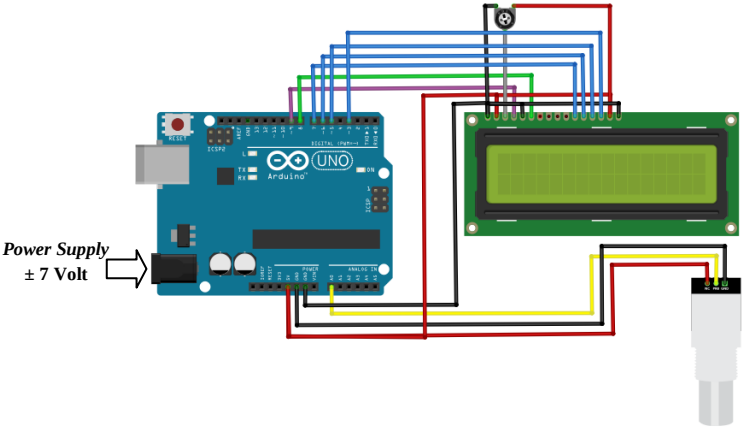
Lalu sebagai alat simulasi, maka akan diibaratkan terdapat 2 titik kebocaran. Dimana letak kebocoran tersebut ditempatkan diantara 3 sensor tekanan. Pada perancangan tersebut, kebocoran pipa digantikan dengan kran air. Jadi ketika salah satu kran air dibuka, maka akan diketahui kebocoran pipa yang dideteksi dari berkurangnya debit air. Titik kebocoran ditentukan sebanyak 2 titik dikarenakan untuk menguji seberapa akurat *prototype* yang akan dibuat.

3.2.2 Rangkaian Sensor Tekanan dan LCD

Pada Gambar 3.4. adalah skematik koneksi Sensor Tekanan ke Arduino.

Gambar 3.4. Koneksi Sensor Tekanan Ke Arduino Uno

Rangkaian Skematik Sensor Tekanan Pin V*out* dari sensor dihubungkan dengan pin A0 Arduino yang dikonfigurasi sebagai *input* yang mana pin A0 tersebut akan memberikan nilai ADC ketika terjadi perubahan nilai tekanan.



Gambar 3.5. Koneksi Sensor Tekanan dan LCD ke Arduino Uno

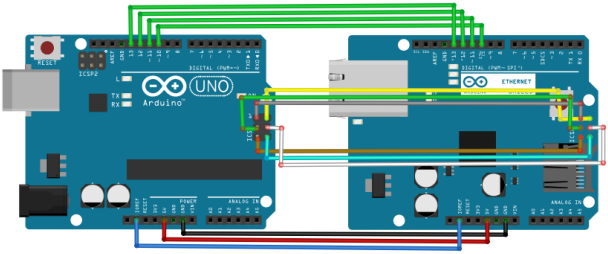
Dari Gambar 3.5 Arduino dibuhungkan pada sensor tekanan dan LCD. Untuk sensor tekanan kabel merah dihubungkan pada pin 5 Volt Arduino. Kabel hitam dihubungkan pada pin Gnd. Dan kabel kuning dihubungkan pada pin A0 Arduino.

Untuk konfigurasi pin LCD, pin Vss dihubungkan pada pin Gnd Arduino. Pin Vdd dihubungkan pada pin Vcc/5 Volt Arduino. Pin V0 LCD dihubungkan pada *multi turn trimpot* yang sudah dihubungkan pada pin 5 Volt dan Gnd. Pin V0 LCD ini gunanya untuk mengatur kecerahan layar LCD. Lalu pin RS pada pin 9 Arduino, pin *Enable* pada pin 8 Arduino, pin DB4, DB5, DB6 dan DB7 masing-masing pada pin 7, 6, 5 dan 3 pin Arduino. Kemudian *Jack* DC disuplai tegangan ± 7 Volt.

3.3 Koneksi *Ethernet Shield* dengan Arduino Uno

Pada perangkat komunikasi digunakan *Ethernet Shield* yang telah *compatible* dengan Arduino Uno. Pemasangan *Ethernet Shield* hanya cukup diletakkan pada bagian atas Arduino Uno. Gabungan antara Arduino Uno dan *Ethernet Shield* dinamakan Arduino *Web Server*. Contoh penggabungan antara *Ethernet shield* dan Arduino Uno dapat dilihat pada Gambar 3.6.

Gambar 3.6. Koneksi *Ethernet Shield* pada Arduino Uno

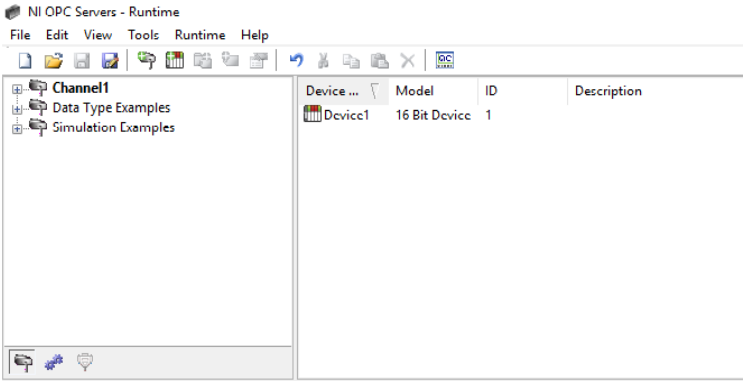


Gambar 3.7. *Schematic Ethernet Shield* pada Arduino Uno

Dari Gambar 3.7 pin 10,11,12 dan 13 adalah masing masing pin *Slave Select*, MOSI, MISO dan SCK yang digunakan untuk terhubung pada *Ethernet Shield.* Pin tersebut tidak bisa digunakan lagi sebagai *input* atau *output* dalam program yang dibuat. Selain pin tersebut masih bisa digunakan langsung dari *Ethernet Shield.*

3.4 Konfigurasi *NI OPC Server*

NI OPC merupakan fasilitas *National Instrument* untuk membuat *server* serta memberikan kemampuan untuk membaca data dan mengirimkanya ke *LabVIEW*. Tugas akhir ini menggunakan *NI OPC Server* untuk menghubungkan *LabVIEW* dengan Arduino Uno. Tanpa *NI OPC* alat ini tidak dapat mengirim dan memmbaca data dari sensor. Berikut gambar tampilan awal dari *NI OPC Server* (Gambar 3.8)

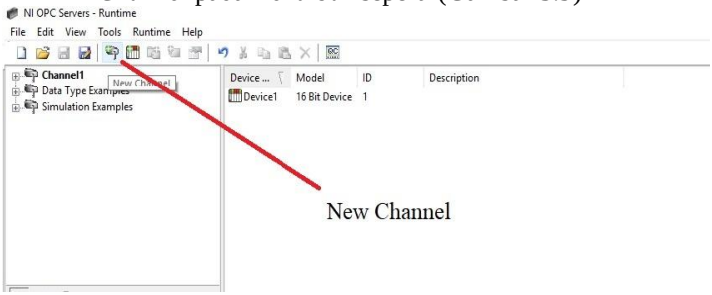


Gambar 3.8. Tampilan *Software NI OPC Server*

3.4.1 Pengaturan *Channel*

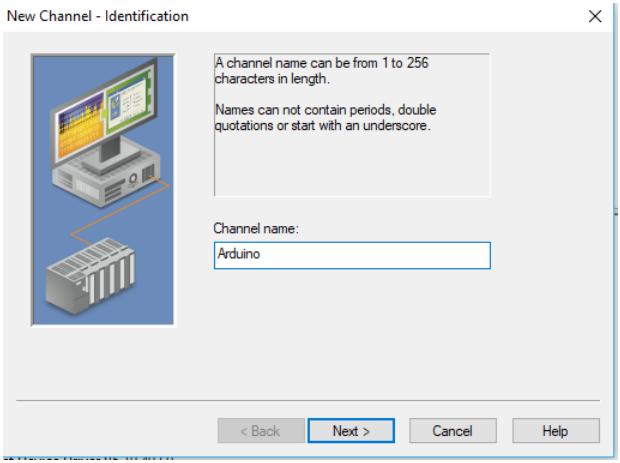
Channel digunakan sebagai sarana *NI OPC* untuk berkomunikasi dengan PC lain atau perangkat keras lain. Sebuah *Channel* digunakan untuk mewakili *networks card* yang terpasang di dalam PC atau *serial port.* Pengaturannya adalah sebagai berikut:

a. Buka *NI OPC Server Configuration*. Setelah itu klik *New Channel* pada *menu bar* seperti (Gambar 3.9)



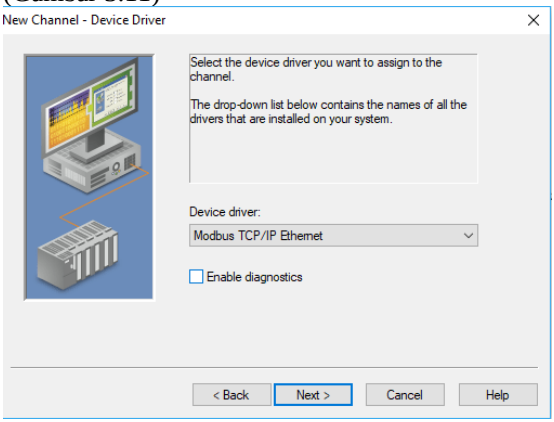
Gambar 3.9. *Setting New Channel*

b. Berikan nama pada perangkat, di sini diberikan nama Wifi 1 (Gambar 3.10)



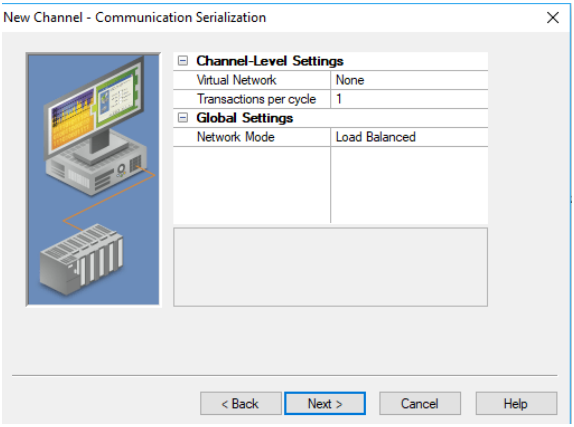
Gambar 3.10. *Setting* Nama *Channel*

c. Setelah itu pilih *Device Driver* sebagai *Modbus TCP/IP Ethernet*. Lalu klik *Next*. Dengan *Enable Diagnostics off* (Gambar 3.11)



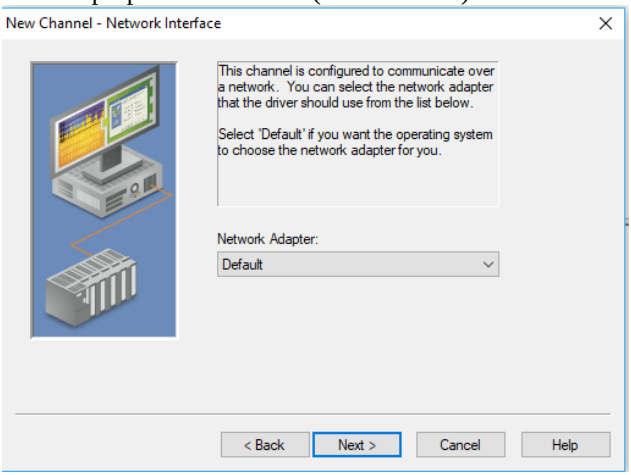
Gambar 3.11. *Setting* Jenis *Channel*

d. Atur *Channel-Level Setting* dan *Global Setting* sepeti Gambar 3.12



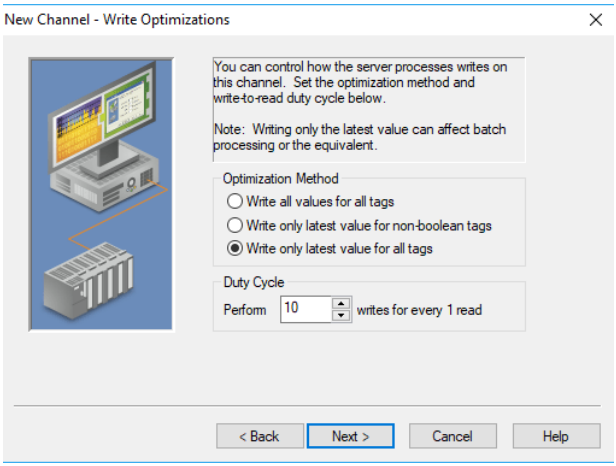
Gambar 3.12. *Setting Level Channel*

e. Atur *network adapter* sesuai dengan *Network Card* yang ada di laptop. Lalu klik *Next* (Gambar 3.13)



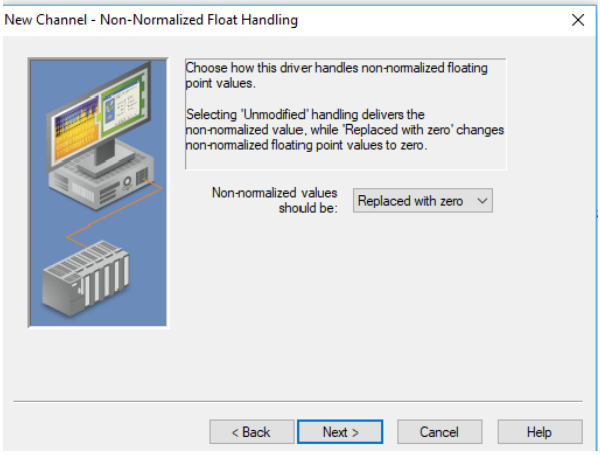
Gambar 3.13. *Setting Network Adapter Channel*

f. Kemudian memilih *optimization setup*, seperti pada Gambar 3.13 lalu klik *Next*.



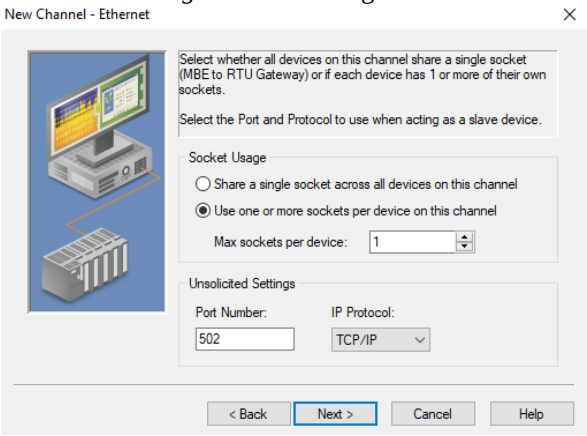
Gambar 3.14. *Setting Optimization Channel*

g. Lalu pilih *non normalized values should be replaced with zero*. Lalu klik *Next*. (Gambar 3.15) seperti



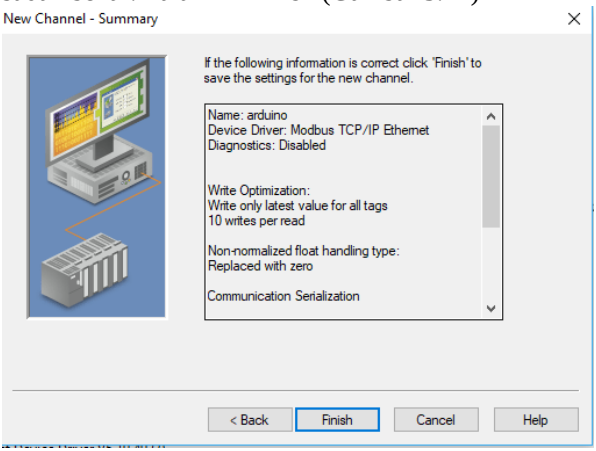
Gambar 3.15. *Setting Non-Normalized Values Channel*

h. Kemudian pilihan *socket usage* pilih *Use one or more per device* dan *Max socket per device* adalah 1. Untuk *unsolicited settings* isi sesuai dengan Gambar 3.16



Gambar 3.16. *Setting Socket Usage Values Channel*

i. Dan terakhir adalah cek semua *setting* yang dilakukan sudah benar. Lalu klik *Finish* (Gambar 3.17)

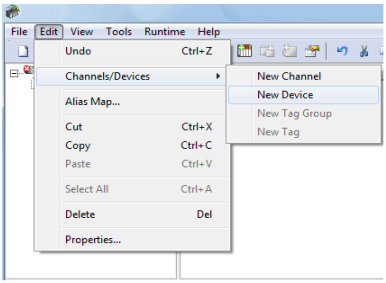


Gambar 3.17 *Setting Summary Channel*

3.4.2 Pengaturan *Device*

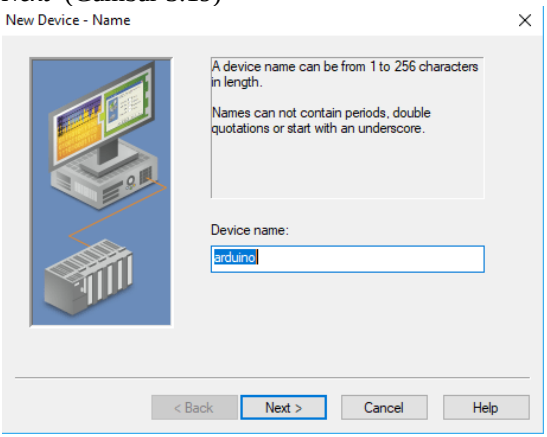
Tahab ini menambahkan *device* pada *channel* yang sudah dibuat. *Device* disini berupa Arduino Uno. Dalam pengaturan ini terdapat *setting* IP yang sudah diprogram di dalam Arduino Uno.

a. Klik *Edit* lalu masuk *menu channels/devices* lalu pilih *new device*



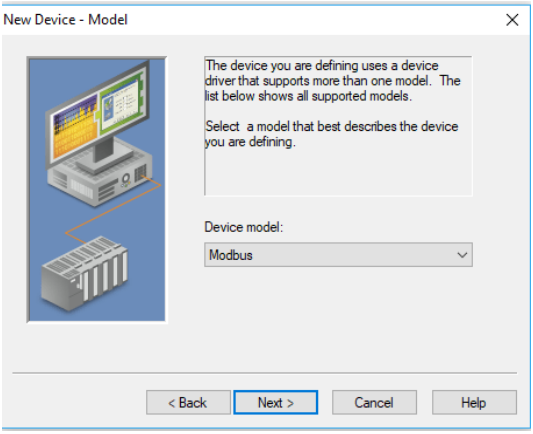
Gambar 3.18. *Setting New Device*

b. Berikan nama pada *device*. Misalkan Arduino. Lalu klik *Next* (Gambar 3.19)



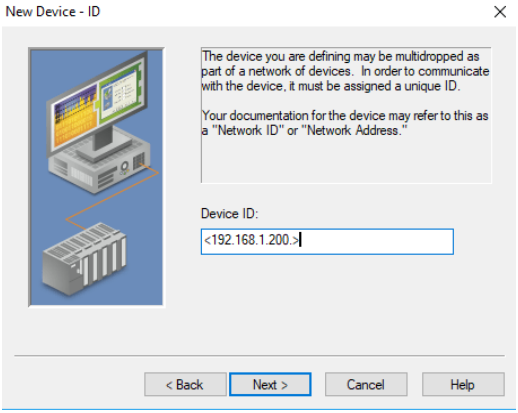
Gambar 3.19. *Setting Name New Device*

c. Lalu pada pilihan *device model*, dan pilihlah *mode modbus.* Setelah itu klik *Next*. (Gambar 3.20)



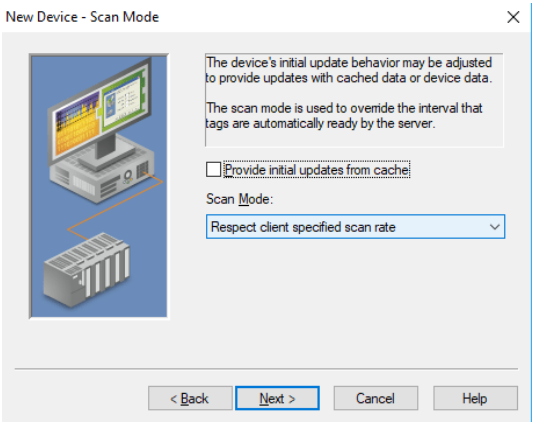
Gambar 3.20. *Setting Model New Device*

d. Kemudian masukkan *Device ID* dengan <192.168.1.XXX>.0 (sesuai dengan program yang di *upload* di Arduino) dan harus memiliki *subnet* yang sama. (Gambar 3.21)



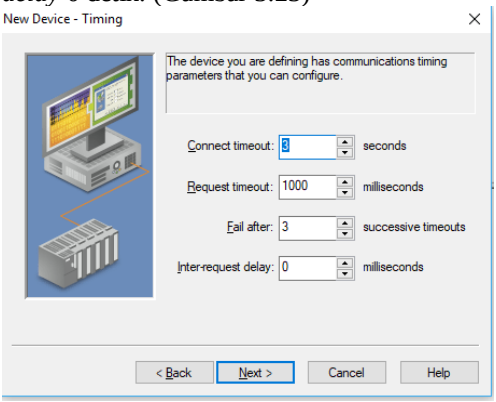
Gambar 3.21. *Setting Network Address New Device*

e. Lalu untuk *scan mode* pilih sesuai dengan Gambar 3.22



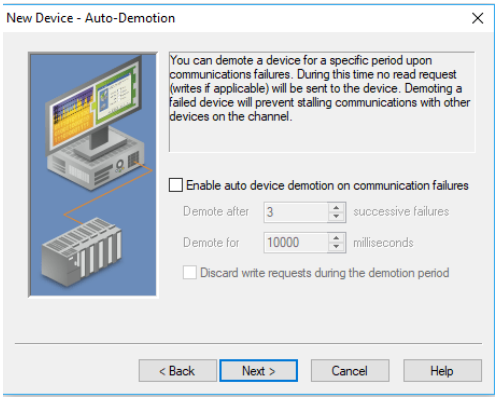
Gambar 3.22. *Setting Scan Mode New Device*

f. Setelah itu tentukan waktu *Connect timeout* 3 detik, *Request timeout* 1 detik, *Fail after* 3 *kali timeout* dan *Inter-quest delay* 0 detik. (Gambar 3.23)



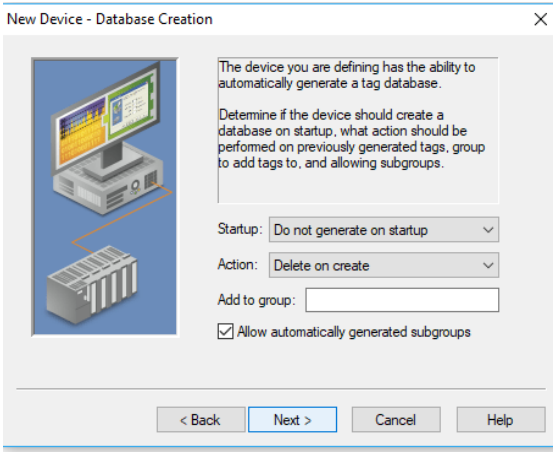
Gambar 3.23. *Setting Timing New Device*

g. Lalu pada pilihan *auto demotion*, pilihan ini bisa dipilih untuk diaktifkan atau tidak, namun pada hal ini *device* di nonaktifkan. (Gambar 3.24)



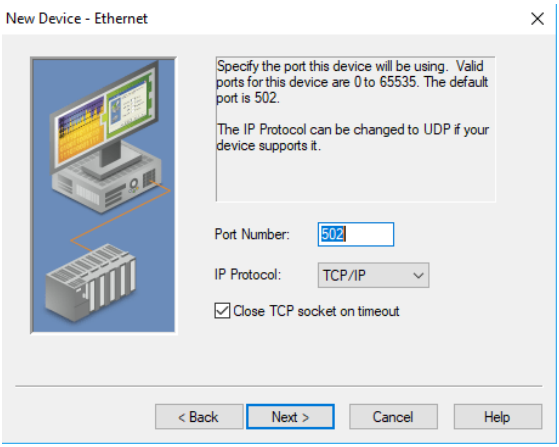
Gambar 3.24. *Setting Auto Demotion New Device*

h. Lalu pada pilihan *database creation*, atur seperti pada Gambar 3.25. Lalu klik *Next*.



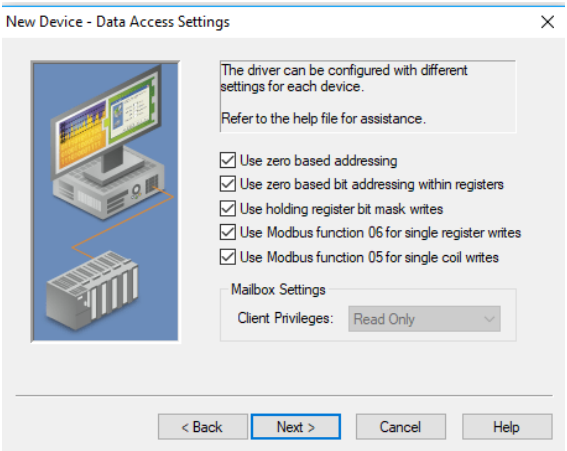
Gambar 3.25. *Setting Database New Device*

i. Kemudian pada *port number* isi dengan 502. IP *Protocol* pilih *TCP/IP*. Kemudian klik *Next*. (Gambar 3.26)



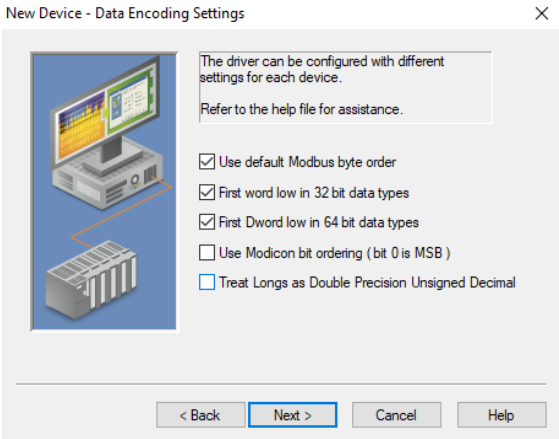
Gambar 3.26. *Setting Ethernet New Device*

j. Lalu pada *Data Acces Settings* centang seperti pada Gambar 3.27. Kemudian klik *Next*.



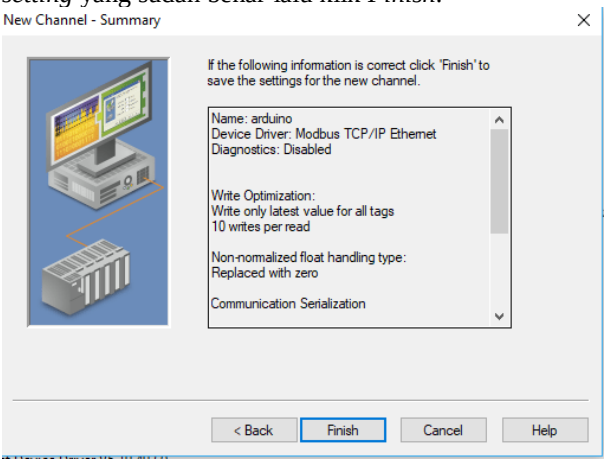
Gambar 3.27. *Setting Data Acces New Device*

k. Setelah pada pengaturan *encoding data system*, cetang seperti Gambar 3.28. Lalu klik *Next*.



Gambar 3.28. *Setting Data Encoding New Device*

l. Langkah yang terakhir adalah *Summary*. Pastikan semua *setting* yang sudah benar lalu klik *Finish*.

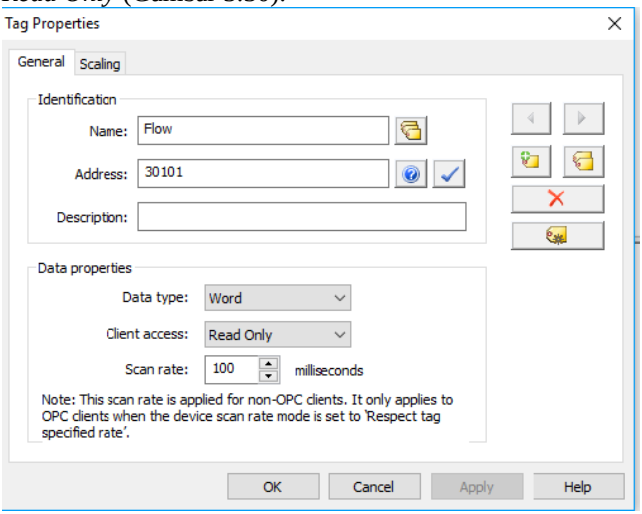


Gambar 3.29. *Summary New Device*

3.4.3 Pengaturan *Tag*

Tag disini berfungsi untuk mewakili alamat data pada *hardware* yang nantinya akan diambil oleh *OPC Server*. Pada pengaturan *tag* kita dapat memmilih tipe data yang akan diambil seperti *word*, *string*, *double*, *boolean* dan sebagainya. Langkah langkah *setting tag* adalah sebagai berikut:

a. Isikan nama *tag* yaitu *Pressure.*
b. Lalu pada alamat tag isikan 30101 (sesuai dengan program Arduino Uno).
c. Pilih *Data Acces* sebagai *word*, lalu *Client Acces* sebagai *Read Only* (Gambar 3.30).



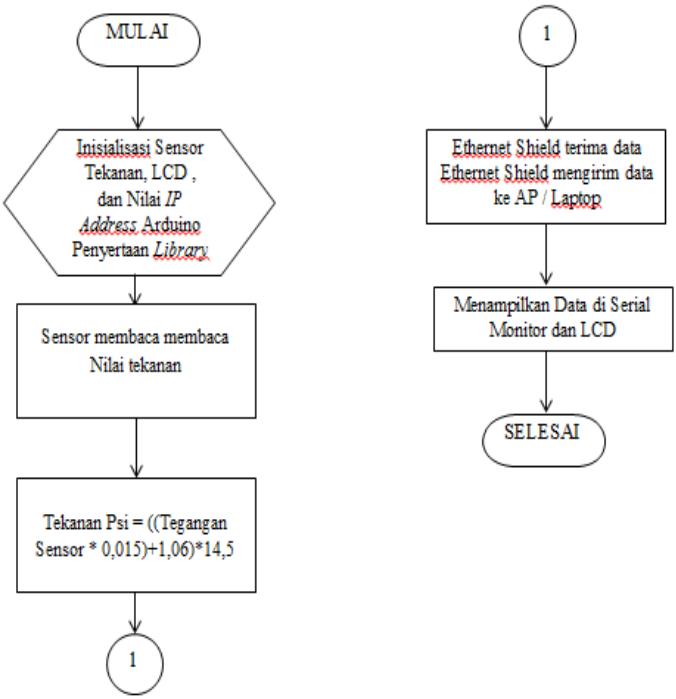
Gambar 3.30. *Setting Model New Device*

3.5 Perancangan Perangkat Lunak (*Software*)

Pada perancangan perangkat *software* pada tugas akhir ini yang dibahas terdiri dari pemrograman pengukuran tekanan oleh sensor pada *software* Arduino IDE, dan perancangan *software LabVIEW.*

3.5.1 Pemrograman Pembacaan Sensor Tekanan

Pemrograman sensor tekanan menggunakaa aplikasi IDE Arduino. 3 sensor tekanan akan membaca tekanan pada 3 titik pada saluran pipa. Untuk sensor tekanan pertama diberi nama P1, sensor kedua diberi nama P2, sensor ketiga diberi nama P3. Selanjutnya dibuat program yang akan membaca tekanan pada 3 sensor. Kemudian program di *upload* ke 3 Arduino Uno yang telah disiapkan. Ketika terjadi perbedaan tekanan pada titik pipa maka akan ditampilkan oleh *LCD*. Data yang didapatkan kemudian ditransimisikan melewati *Ethernet Shield* dan *Acces Point* yang nantinya akan ditampilkan pada *LabVIEW* Berikut ini adalah *flowchart* program Arduino (Gambar 3.31). Program keseluruhan dapat dilihat pada Lampiran A.

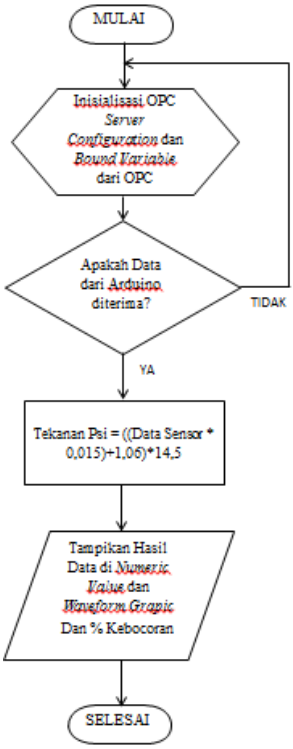


Gambar 3.31. *Flowchart* Program Arduino Uno

Tahap pertama dari *flowchart* (Gambar 3.30) tersebut adalah inisialisasi *Ethernet*, SPI, Modbus dan LCD. Setelah inisialisasi selanjunya jika sensor membaca tekanan akan dilanjutkan pada rumus Tekanan Psi = ((Tegangan Sensor * 0,0015)+1.06)*14,5. Selanjutnya *Ethernet Shield* akan mengirim data ke Laptop melalui *Acces Point*. Selanjutnya data akan ditampilkan di *serial monitor* dan LCD.

3.5.2 Perancangan Software *LabVIEW*

Pada perancangan *software LabVIEW* yang pertama adalah membuat *flowchart software LabVIEW*. *Flowchat* dapat dilihat pada Gambar 3.32

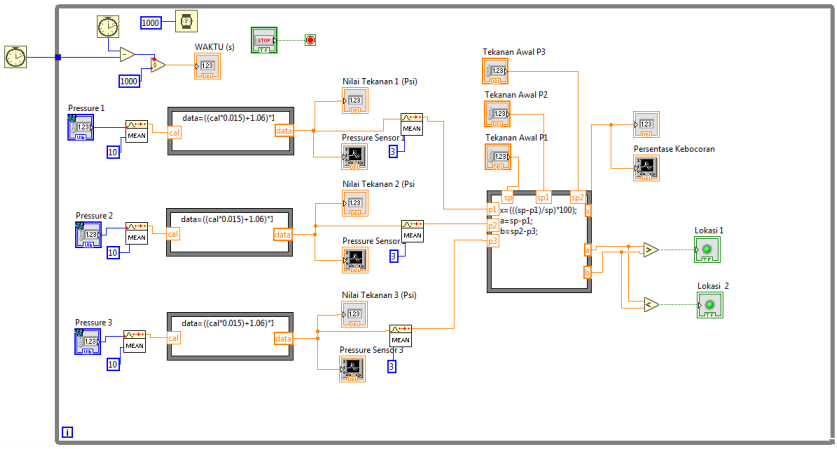


Gambar 3.32. *Flowchart* Program *LabVIEW*

Dari *flowchart* program *LabVIEW* kemudian disususn *block diagram*. *Block diagram* berisi *source code* yang berfungsi sebagai instruksi untuk *front panel*. Sedangkan pada *front panel* sendiri mengandung *control* dan indikator untuk membangun sebuah VI (*Virtual Instruments*), menjalankan program dan men*debug* program.

Pada perancangan *block diagram* (Gambar 3.33) bagian kiri terdapat *function palatte numeric* dengan tanda nama Data1, Data 2 dan Data 3 yang berguna untuk menampilkan pembacaan 3 sensor tekanan tiap 2 detik sekali, ditunjukkan pada *function palatte* dengan *icon* jam per 200 ms, ditunjukkan pada *function palatte* dengan *icon* jam, yang dikirim melalui *Ethernet*. Pada bagian kanan Data, Data 2 dan Data 3 terdapat kotak berwarna abu-abu yang disebut dengan *formula node*, fungsi dari *function palatte* ini adalah untuk menuliskan rumus konversi sinyal *digital* menjadi besaran tekanan (Psi). Lalu pada bagian sebelah kanan *formula node* disambungkan pada *function palatte numeric* dan grafik sebagai hasil perhitungan tekanan air dengan tanda nama *Pressure 1, Pressure 2,* dan *Pressure 3* untuk grafik dan Nilai Tekanan 1 (Psi), Nilai Tekanan 2 (Psi), nilai Tekanan 3 (Psi) untuk *numeric indicator.*

Nilai yang didapatkan dari hasil perhitungan lalu diambil 5 sampel data untuk diambil rata-rata. Setelah mendapatkan nilai rata-rata 5 sampel data maka akan dilakukan perhitungan persentase kebocoran pada *function palatte formula node*. Setelah didapatkan nilai perhitungan persentase kebocoran, disambungkan pada *function palatte for loop* yang didalamnya terdapat *function palatte numeric*, grafik, dan *wait* untuk menampilkan nilai perhitungan dan grafik per 2000 ms.



Gambar 3.33. Rancangan *Block Diagram* LabVIEW

Selanjutnya yaitu membuat tampilan *front panel*. Cara memasukkan gambar pada *front panel LabVIEW* prinsipnya tinggal *drag and drop*. Untuk mendapatkan gambar yang disediakan oleh *LabVIEW* caranya adalah :

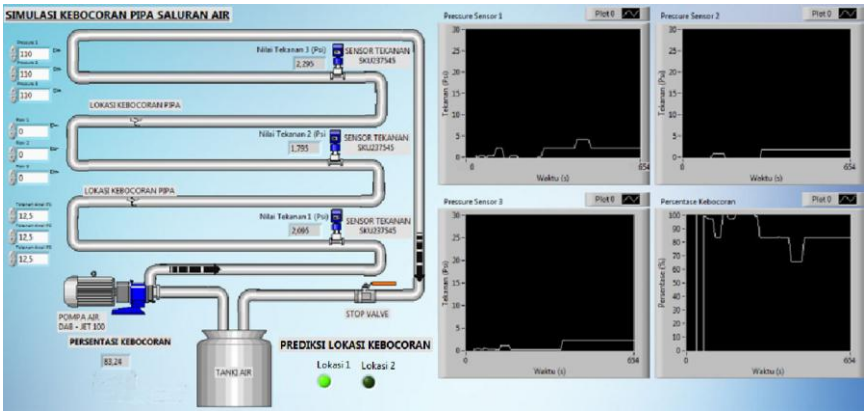
1. Pada *front panel LabVIEW* klik *menu bar "Tools"*.
2. Lalu pindahkan kurusor ke menu *"DCS Module"*.
3. Lalu klik "*Image Navigator"*.
4. Pada menu "*Image Navigator"* pilih gambar lalu *drag and drop* gambar yang dibutuhkan.

Perancangan pada *front panel* pada Gambar 3.34 dibuat gambar instalasi pipa yang dilengkapi dengan sensor tekanan dimana pada masing-masing sensor terdapat keterangan diatasnya. Pada bagian bawah tanda nama Data1, Data 2, dan Data 3 merupakan hasil dari pembacaan sensor.

Lalu pada bagian bawah tanda nama *Pressure 1*, *Pressure 2* dan *Pressure 3* untuk grafik dan Nilai Tekanan 1 (Psi), Nilai Tekanan 2 (Psi), Nilai Tekanan 3 (Psi), untuk *numeric indicator*. Nilai Tekanan (Psi) merupakan hasil dari perhitungan konversi sinyal *analog* menjadi besaran tekanan dengan Psi.

Hasil perhitungan persentase kebocoran ditampilakan pada bagian kanan. Sedangkan gambar grafik diletakkan pada bagian

kanan secara *parallel*, yang dimana telah terdapat tanda nama pada bagian atas gambar grafik.



Gambar 3.34. Gambar Rancangan *Front Panel LabVIEW*

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

BAB IV
PENGUJIAN DAN ANALISA ALAT

Dalam membuat suatu sistem, pengujian dan analisa sangat diperlukan. Pengujian dan analisa bertujuan untuk mengetahui system bekerja sesuai rencana atau belum. Dari hasil pengujian dan analisadapat diketahui kelemahan-kelemahan dari sistem, sehingga dapat dilakukan perbaikan, pengembangan, dan penyempurnaan sistem.

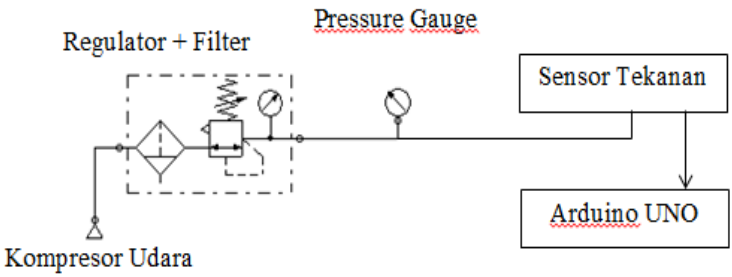
Dalam bab ini dibahas pengujian serta analisa dari perancangan sistem yang telah dibuat pada tugas akhir. Pengujian dilakukan secara parsial terlebih dahulu, tujuannya adalah untuk mengetahui kinerja *hardware* setiap unit. Setelah mengetahui respon kinerja *hardware* setiap unit, kemudian akan dilakukan pengujian integrasi sistem secara keseluruhan. Adapun beberapa pengujian yang telah dilakukan adalah sebagai berikut.

4.1 Kalibrasi Sensor Tekanan

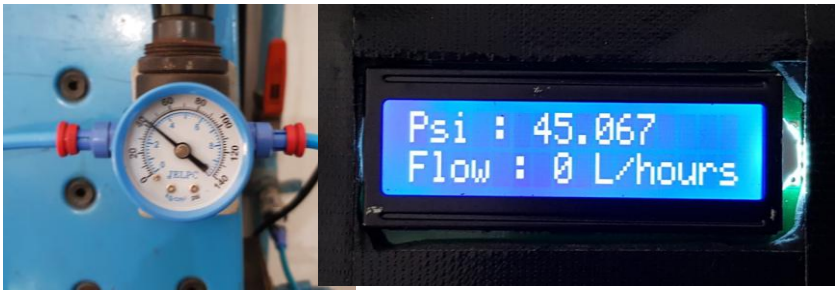
Sensor Tekanan yang disudah diberi program akan diuji hasil pembacaan tekanan dengan *Pressure Gauge*. Sebelum dilakukan pengukuran kebocoran pada instalasi pipa, maka dibutuhkan kalibrasi sensor untuk menentukan keakuratan hasil pembacaan alat ini. Sensor ini dikalibrasi menggunakan *Pressure Gauge JELPC* di Bengkel Pneumatik BLK. Langkah-langkah kalibrasi sensor tekanan adalah sebagai berikut:

1. Menyiapkan peralatan yang dibutuhkan meliputi sensor tekanan, Arduino Uno, *Multimeter*, selang 6mm, *connector* 6mm to ½ dim, selotip pipa dan kabel *jumper*.
2. Menyambungkan sensor ke Arduino Uno yang telah di program pada pin A0, Vcc dan Gnd.
3. Menyambungkan kabel USB Arduino Uno ke *port* USB laptop.
4. Memasangkan *connector* 6mm ke ½ dim di sensor tekanan menggunakan selotip pipa.
5. Menyambungkan selang 6mm ke *Pressure Gauge JELPC* dilanjutkan ke Sensor Tekanan 1 seperti Gambar 4.1
6. Menyalakan kompresor udara dan buka *valve* udara.

7. Memutar *Pressure Gauge* searah jarum jam atur hingga tekanan menunjuk 0 Kg/cm^2, 1 Kg/cm^2, 1,5 Kg/cm^2, 2 Kg/cm^2, 2,5 Kg/cm^2, 3 Kg/cm^2, 3,5 Kg/cm^2, 4 Kg/cm^2, 4,5 Kg/cm^2, 5 Kg/cm^2. Dimana 1 Kg/cm^{2}adalah 14,223 Psi dan tekanan maksimal kompresor udara adalah 6 Kg/cm^2.
8. Mencatat nilai tekanan (Psi) pada LCD bandingkan dengan nilai tekanan yang tertera pada *Pressure Gauge* (Gambar 4.2)
9. Mengulangi langkah 1-8 untuk Sensor 2, dan 3.



Gambar 4.1 Sambungan *Pressure Gauge* dan Sensor Tekanan



Gambar 4.2 Proses Kalibrasi Sensor Tekanan

Tabel 4.1 Hasil Kalibrasi Sensor 1

Tekanan *Pressure Gauge*		Tekanan Sensor 1 (Psi)	*Error* (%)
Tekanan (Kg/cm^2)	Tekanan (Psi)		
0	0	0,048	-
1	14,223	14,83	4,267735358
1,5	21,334	22,44	5,181747873
2	28,446	29,84	4,900513253
2,5	35,557	36,36	2,256907825
3	42,669	45,06	5,603599803
3,5	49,780	53,54	7,552153956
4	56,892	61,83	8,679603459
4,5	64,003	68,98	7,775356035
5	71,115	75,51	6,180130774

Berdasarkan hasil pengukuran kalibrasi Tabel 4.1 sensor tekanan dengan *Pressure Gauge* hampir menunjukkan angka yang sama. Data yang diperoleh dari keseluruhan sensor menunjukkan hasil yang *linier* dan stabil. Hasil pengukuran didapatkan *error* dibawah angka 9%. Persentase *error* terbesar terjadi ketika pengukuran 56,89 Psi dimana alat membaca 61,83 Psi

Tabel 4.2 Hasil Kalibrasi Sensor 2

Tekanan *Pressure Gauge*		Tekanan Sensor 2 (Psi)	*Error* (%)
Tekanan (Kg/cm^2)	Tekanan (Psi)		
0	0	0,2693	-
1	14,223	14,393	1,195247135
1,5	21,334	22,443	5,195809604
2	28,446	30,493	7,196090839
2,5	35,557	37,453	5,330802222
3	42,669	45,933	7,649581664
3,5	49,780	53,117	6,702423646
4	56,892	60,727	6,740842298
4,5	64,003	68,777	7,458185887
5	71,115	76,167	7,103986501

Berdasarkan hasil pengukuran kalibrasi Tabel 4.2 sensor tekanan dengan *Pressure Gauge* memiliki selisih hasil pembacaan yang masih dapat ditoleransi. Data yang diperoleh dari keseluruhan sensor menunjukkan hasil yang *linier* dan stabil. Hasil pengukuran didapatkan *error* dibawah angka 8%.

Tabel 4.3 Hasil Kalibrasi Sensor 3

Tekanan *Pressure Gauge*		Tekanan Sensor 3 (Psi)	*Error* (%)
Tekanan (Kg/cm^2)	Tekanan (Psi)		
0	0	-0,187	-
1	14,223	13,527	4,893482388
1,5	21,334	22,227	4,183364972
2	28,446	30,057	5,663362167
2,5	35,557	37,017	4,104619279
3	42,669	45,067	5,620005156
3,5	49,780	53,987	8,450095921
4	56,892	60,727	6,740842298
4,5	64,003	68,557	7,114454678
5	71,115	76,387	7,413344583

Berdasarkan hasil pengukuran kalibrasi Tabel 4.3 sensor tekanan dengan *Pressure Gauge* hampir menunjukkan angka yang sama. Data yang diperoleh dari keseluruhan sensor menunjukkan hasil yang *linier* dan stabil. Hasil pengukuran didapatkan *error* dibawah angka 8,5%. Untuk pembacaan tekanan 0 Psi alat membaca -0,187 Psi dan pembacaan ketika 49,78 Psi memiliki persentase kesalahan terbesar.

4.2 Pengujian Komunikasi *Ethernet*

Pengujian ini dilakukan pada modul *Ethernet shield* yang telah dipasang pada bagian atas *board* Arduino Uno. Dalam hal ini pengujian komunikasi dibagi menjadi dua, yaitu pengujian komunikasi *Ethernet* pada *board* Arduino Uno dan pengujian komunikasi *Ethernet* dengan *software LabVIEW*.

Sebelum menggunakan komunikasi *Ethernet*, sebaiknya dilakukan pengujian komunikasi *Ethernet* pada *board* Arduino Uno terlebih dahulu. Pengujian ini dilakukan dengan cara menyamakan

setting IP (*Internet Protocol*) *address* yang telah diprogram pada *board* Arduino Uno, dapat dilihat pada Gambar 4.3, Gambar 4.5, dan Gambar 4.7, dengan *setting* IP *address* pada *Ethernet shieild*. *Setting* IP *address* pada *Ethernet shield* dapat diketahui melalui *command prompt* pada suatu *notebook*/komputer. (Gambar 4.4, Gambar 4.6, dan Gambar 4.8)

```
void setup()
{
    lcd.begin(16, 2);
    byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
    byte ip[] = { 192, 168, 1, 196 };
    mb.config(mac, ip);
    pinMode(psensor, INPUT);
    Serial.begin(9600);
    attachInterrupt(0, rpm, RISING);
    mb.addIreg(sencalc);
    mb.addIreg(water);
    sei();
}
```

Gambar 4.3. Program pada Arduino Uno ke-1

```
C:\Users\Aspire V5-471>ping 192.168.1.196

Pinging 192.168.1.196 with 32 bytes of data:
Reply from 192.168.1.196: bytes=32 time<1ms TTL=128
Reply from 192.168.1.196: bytes=32 time<1ms TTL=128
Reply from 192.168.1.196: bytes=32 time<1ms TTL=128
Reply from 192.168.1.196: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.196:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Aspire V5-471>
```

Gambar 4.4. *Command Prompt* pada Laptop Dengan IP *address* Arduino Uno ke-1

```
void setup()
{
    lcd.begin(16, 2);
    byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
    byte ip[] = { 192, 168, 1, 198 };
    mb.config(mac, ip);
    pinMode(psensor, INPUT);
    Serial.begin(9600);
    attachInterrupt(0, rpm, RISING);
    mb.addIreg(sencalc);
    mb.addIreg(water);
    sei();
}
```

Gambar 4.5. Program pada Arduino Uno ke-2

```
C:\Users\Aspire V5-471>ping 192.168.1.198

Pinging 192.168.1.198 with 32 bytes of data:
Reply from 192.168.1.198: bytes=32 time=4ms TTL=128
Reply from 192.168.1.198: bytes=32 time=2ms TTL=128
Reply from 192.168.1.198: bytes=32 time=2ms TTL=128
Reply from 192.168.1.198: bytes=32 time=4ms TTL=128

Ping statistics for 192.168.1.198:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 4ms, Average = 3ms

C:\Users\Aspire V5-471>ping _
```

Gambar 4.6. *Command Prompt* pada Laptop Dengan IP *address* Arduino Uno ke-2

```
void setup()
{
    lcd.begin(16, 2);
    byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
    byte ip[] = { 192, 168, 1, 200 };
    mb.config(mac, ip);
    pinMode(psensor, INPUT);
    Serial.begin(9600);
    attachInterrupt(0, rpm, RISING);
    mb.addIreg(sencalc);
    mb.addIreg(water);
    sei();
}
```

Gambar 4.7. Program pada Arduino Uno ke-3

```
C:\Users\Aspire V5-471>ping 192.168.1.200

Pinging 192.168.1.200 with 32 bytes of data:
Reply from 192.168.1.200: bytes=32 time=7ms TTL=128
Reply from 192.168.1.200: bytes=32 time=3ms TTL=128
Reply from 192.168.1.200: bytes=32 time=5ms TTL=128
Reply from 192.168.1.200: bytes=32 time=2ms TTL=128

Ping statistics for 192.168.1.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 7ms, Average = 4ms

C:\Users\Aspire V5-471>_
```

Gambar 4.8. *Command Prompt* pada Laptop Dengan IP *address* Arduino Uno ke-3

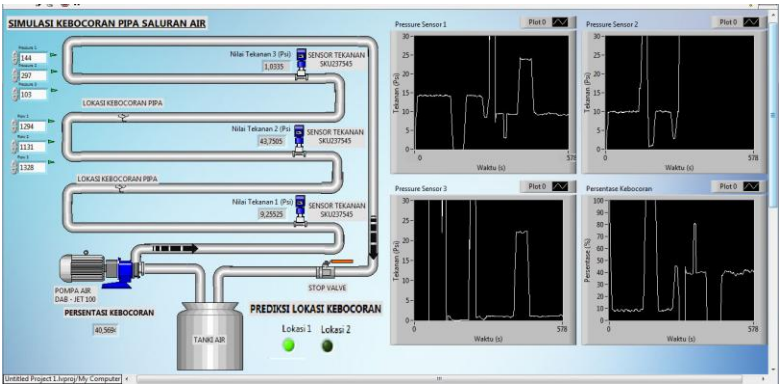
Dapat dilihat pada gambar diatas menunjukkan bahwa *Ethernet shield* yang tersambung pada laptop telah siap digunakan. Ketika

mengirim perintah *ping* nomor IP *address* pada kolom *command prompt* maka laptop akan memunculkan informasi bahwa nomor IP *address* telah bekerja, dapat dilihat pada Gambar 4.4, Gambar 4.6 dan Gambar 4.8.

4.3 Pengujian pada *Software LabVIEW*

Setelah pengujian komunikasi *Ethernet* pada Arduino Uno, pengujian komunikasi *Ethernet* perlu dilakukan pada *software LabVIEW* untuk mengetahui apakah *Ethernet shield* dapat mengirimkan data pada *software LabVIEW* atau tidak.

Pengujian ini dilakukan dengan cara parsial pada rangkaian sensor tekanan ke-1 sampai sensor ke-3. Setelah rangkaian sensor tersambung, alirkan air pada sensor. Sembari mengalirkan air pada sensor, amati tampilan pada *software LabVIEW*. Setelah itu catat nilai yang terbaca pada tampilan *LabVIEW*.



Gambar 4.9. Tampilan *LabVIEW* dalam Keadaan *Run*

Dapat dilihat pada Gambar 4.9 diketahui bahwa *setting* yang sudah dilakukan berjalan dan data dari Arduino dapat dikirim ke *LabVIEW* ditampilkan dalam bentuk *pallete numeric*. Dimana ketika *pallete numeric* tersebut bisa terhubung pada Arduino, maka panah akan berwarna hijau dan terdapat keterangan *Connected to Pressure*.

4.4 Pengujian Alat Simulasi

Pengujian secara keseluruhan merupakan gabungan dari beberapa pengujian yang telah dilakukan dan pengujian kebocoran pipa pada *prototype* simulasi distribusi air

4.4.1 Pengujian Tanpa Aliran

Letak titik kebocoran dapat dilihat pada Gambar 3.1. Untuk pengujian tanpa aliran, yang pertama dilakukan adalah membaca nilai tekanan yang tertera pada LCD pada kondisi pompa mati Langkah langkah dari pengujian ini antara lain:

1. Memasang semua sensor pada titik yang sudah ada di pipa.
2. Menyalakan Arduino dan sambungkan sensor pada Arduino Uno.
3. Mencatat nilai yang ditampilkan di LCD titik 1.
4. Mengulangi langkah 1-3 untuk titik 2 dan 3

Hasil pengambilan data adalah sebagai berikut

Tabel 4.4. Data Pengujian Alat Pompa *Off*

No.	Kondisi	Titik 1 (Psi)	Titik 2 (Psi)	Titik 3 (Psi)
1	Tanpa Aliran(Pompa OFF)	0,0486	0,260	-0,40
2		0,0486	-0,40	-0,18
3		00486	0,260	-0,40

Pada Tabel 4.4 keadaan alat simulasi pompa dalam keadaan off, semua kebocoran ditutup dan *stop valve* dibuka 90^0 (dibuka penuh). Dari Tabel 4.4 dapat disimpulkan bahwa pembacaan 3 sensor tekanan masih akurat dengan selisih paling besar ±0,5 Psi di keadaan tanpa aliran (pompa air *off*)

4.4.2 Pengujian *Stop Valve*

Letak titik kebocoran dapat dilihat pada Gambar 3.1. Untuk pengujian *stop valve*, langkah langkah dari pengujian ini antara lain:

1. Memasang semua sensor pada titik yang sudah ada di pipa.
2. Menyalakan Arduino dan sambungkan sensor pada Arduino Uno.

3. Menyalakan pompa dan atur *stop valve* pada $22,5^0$, 45^0, 54^0, 63^0 dan 72^0.
4. *Stop Valve* dapat dibuka dari nilai $0^0 – 90^0$.
5. Mencatat nilai yang ditampilkan di LCD titik 1.
6. Mengulangi langkah 1-4 untuk titik 2 dan 3.

Tabel 4.5 Tekanan Ketika *Stop Valve* Ditutup $\pm 22^0$

No.	Tekanan Titik 1 (Psi)	Tekanan Titik 2 (Psi)	Tekanan Titik 3 (Psi)
Percobaan 1	4,39	3,95	2,87
Percobaan 2	4,60	3,52	2,65
Percobaan 3	4,39	3,95	2,87

Dari Tabel 4.5 dapat dilihat bahwa pada saat *stop valve* ditutup $\pm 22^0$ maka tekanan dalam pipa meningkat ±4 Psi. Dari tabel juga dapat disimpulkan bahwa pembacaan ketiga sensor mendekati mempunyai *range* antara 2,8-4,6 Psi.

Tabel 4.6 Tekanan Ketika *Stop Valve* Ditutup $\pm 45^0$

No.	Tekanan Titik 1 (Psi)	Tekanan Titik 2 (Psi)	Tekanan Titik 3 (Psi)
Percobaan 1	9,71	8,74	7,65
Percobaan 2	9,39	8,52	7,43
Percobaan 3	9,39	8,95	7,65

Dari Tabel 4.6 dapat dilihat bahwa pada saat *stop valve* ditutup $\pm 45^0$ maka tekanan dalam pipa meningkat ±10 Psi. Dari tabel juga dapat disimpulkan bahwa pembacaan ketiga sensor mendekati mempunyai *range* antara 7,6-9,7 Psi.

Tabel 4.7 Tekanan Ketika *Stop Valve* Ditutup $\pm 54^0$

No.	Tekanan Titik 1 (Psi)	Tekanan Titik 2 (Psi)	Tekanan Titik 3 (Psi)
Percobaan 1	11,78	11,78	10,69
Percobaan 2	12,00	12,22	10,48
Percobaan 3	11,13	11,56	10,26

Dari Tabel 4.7 dapat dilihat bahwa pada saat *stop valve* ditutup $\pm 54^0$ maka tekanan dalam pipa meningkat ±12 Psi. Dari tabel juga

dapat disimpulkan bahwa pembacaan ketiga sensor mendekati mempunyai *range* antara 10,2-12,0 Psi.

Tabel 4.8 Tekanan Ketika *Stop Valve* Ditutup $\pm 63^0$

No.	Tekanan Titik 1 (Psi)	Tekanan Titik 2 (Psi)	Tekanan Titik 3 (Psi)
Percobaan 1	15,70	15,70	14,17
Percobaan 2	15,26	15,26	14,39
Percobaan 3	15,48	15,70	14,83

Dari Tabel 4.8 dapat dilihat bahwa pada saat *stop valve* ditutup $\pm 63^0$ maka tekanan dalam pipa meningkat ±15 Psi. Dari tabel juga dapat disimpulkan bahwa pembacaan ketiga sensor mendekati mempunyai *range* antara 14,1-15,7 Psi.

Tabel 4.9 Tekanan Ketika *Stop Valve* Ditutup $\pm 72^0$

No.	Tekanan Titik 1 (Psi)	Tekanan Titik 2 (Psi)	Tekanan Titik 3 (Psi)
Percobaan 1	24,18	23,53	23,75
Percobaan 2	23,31	24,83	23,31
Percobaan 3	24,18	23,40	23,53

Dari Tabel 4.9 dapat dilihat bahwa pada saat *stop valve* ditutup $\pm 72^0$ maka tekanan dalam pipa meningkat ±24 Psi. Dari tabel juga dapat disimpulkan bahwa pembacaan ketiga sensor mendekati mempunyai *range* antara 23,3-24,2 Psi.

4.4.3 Pengujian Kebocoran

Pada subbab ini dilakukan pengujian terhadap alat simulasi ketika keadaan kebocoran 1 dan kebocoran 2 dibuka. Langkah langkah dari pengujian ini adalah sebagai berikut:

1. Memasang semua sensor pada titik yang sudah ada di pipa.
2. Menyalakan Arduino dan sambungkan sensor pada Arduino Uno
3. Menyalakan pompa air dan atur *stop valve* pada $\pm 54^0$ tertutup
4. Mencatat tekanan yang ditampilkan pada LCD dengan kondisi awal, kondisi kran kebocoran 1 dibuka 20^0, kondisi

kran kebocoran 2 dibuka 20^0, kondisi kran kebocoran 1 dibuka 30^0 dan kondisi kran kebocoran 2 dibuka 30^0.

5. Mengulangi langkah 1-4 untuk titik 2 dan 3

Data hasil pengujian adalah sebagai berikut :

Tabel 4.10 Tekanan Ketika *Stop Valve* Ditutup $\pm 54^0$ Tanpa Kebocoran (Normal)

No.	Tekanan Titik 1 (Psi)	Tekanan Titik 2 (Psi)	Tekanan Titik 3 (Psi)
1	12,72	11,72	9,98
2	12,70	11,85	9,69
3	12,78	11,67	9,56
4	12,72	11,91	9,98
5	12,76	11,78	10,04

Dari Tabel 4.10 dapat dilihat bahwa pada saat *stop valve* ditutup $\pm 54^0$ dan titik kebocoran 1 dan 2 ditutup, tekanan yang pada titik 1, 2 dan 3 adalah seperti pada Tabel 4.10.

Tabel 4.11 Tekanan Ketika *Stop Valve* Ditutup $\pm 54^0$ Kran Kebocoran 1 Dibuka $\pm 20^0$

No.	Tekanan Titik 1 (Psi)	Tekanan Titik 2 (Psi)	Tekanan Titik 3 (Psi)
1	11,06	10,06	8,48
2	10,93	10,13	8,45
3	10,87	9,98	8,45
4	11,04	10,13	8,52
5	10,91	10,13	8,52

Dari Tabel 4.11 dapat dilihat bahwa pada saat *stop valve* ditutup $\pm 54^0$ dan titik kebocoran 1 dibuka sebesar $\pm 20^0$ dan titik kebocoran 2 ditutup, tekanan yang pada titik 1, 2 dan 3 adalah seperti pada Tabel 4.11

Tabel 4.12 Tekanan Ketika *Stop Valve* Ditutup ±54^0 Kran Kebocoran 2 Dibuka ±20^0

No.	Tekanan Titik 1 (Psi)	Tekanan Titik 2 (Psi)	Tekanan Titik 3 (Psi)
Percobaan 1	10,45	9,37	7,89
Percobaan 2	10,65	9,39	7,98
Percobaan 3	10,50	9,46	8,04
Percobaan 4	10,41	9,48	8,02
Percobaan 5	10,63	9,54	7,98

Dari Tabel 4.12 dapat dilihat bahwa pada saat *stop valve* ditutup ±54^0 dan titik kebocoran 2 dibuka sebesar ±20^0 dan titik kebocoran 1 ditutup, tekanan yang pada titik 1, 2 dan 3 adalah seperti pada Tabel 4.12.

Tabel 4.13 Tekanan Ketika *Stop Valve* Ditutup ±54^0 Kran Kebocoran 1 Dibuka ±30^0

No.	Tekanan Titik 1 (Psi)	Tekanan Titik 2 (Psi)	Tekanan Titik 3 (Psi)
Percobaan 1	6,56	5,69	4,65
Percobaan 2	6,54	5,63	4,60
Percobaan 3	6,52	5,65	4,58
Percobaan 4	6,48	5,69	4,65
Percobaan 5	6,63	5,65	4,58

Dari Tabel 4.13 dapat dilihat bahwa pada saat *stop valve* ditutup ±54^0 dan titik kebocoran 1 dibuka sebesar ±30^0 dan titik kebocoran 2 ditutup, tekanan yang pada titik 1, 2 dan 3 adalah seperti pada Tabel 4.13.

Tabel 4.14 Tekanan Ketika *Stop Valve* Ditutup ±54^0 Kran Kebocoran 2 Dibuka ±30^0

No.	Tekanan Titik 1 (Psi)	Tekanan Titik 2 (Psi)	Tekanan Titik 3 (Psi)
Percobaan 1	6,37	5,28	4,13
Percobaan 2	6,30	5,42	4,19
Percobaan 3	6,43	5,21	4,13
Percobaan 4	6,32	5,28	4,06
Percobaan 5	6,48	5,34	4,13

Dari Tabel 4.14 dapat dilihat bahwa pada saat *stop valve* ditutup ±54^0 dan titik kebocoran 1 dibuka sebesar ±30^0 dan titik kebocoran 2 ditutup, tekanan yang pada titik 1, 2 dan 3 adalah seperti pada Tabel 4.14.

4.5 Pengujian Data *Telemetring LabVIEW*

Pengujian *Telemetring* ini merupakan pengujian dari data yang ditampilkan pada *software LabVIEW*. Dimana pada titik 1 data ditransmisikan melalui kabel UTP (RJ-45). Di titik 2 data ditransmisikan melalui *Acces Point* kemudian dilanjutkan ke *LabVIEW*. Di titik 3 data ditransmisikan melalui *Acces Point* kemudian dilanjutkan ke *LabVIEW*.

Tabel 4.15 Tekanan Ketika *Stop Valve* Ditutup ±54^0

No.	Titik 1 (Psi)	Titik 2 (Psi)	Titik 3 (Psi)
1	13,3397	12,8875	12,4744
2	13,0135	12,8445	12,4745
3	12,9767	12,8222	12,4305
4	12,9917	12,8875	12,6915
5	13,1657	12,8657	12,4958
6	12,8177	12,844	12,5175
7	12,9482	12,6482	12,3653
8	12,9047	12,6917	12,3653
9	12,9767	12,6482	12,1913

No.	Titik 1 (Psi)	Titik 2 (Psi)	Titik 3 (Psi)
10	12,9047	12,6047	12,4305
Set Point	13,0025	12,7743	12,4435

Dari Tabel 4.15 dapat dilihat bahwa pada saat *stop valve* ditutup $\pm 54^0$ dan titik kebocoran 1 dan 2 ditutup, tekanan yang pada titik 1, 2 dan 3 adalah seperti pada Tabel 4.15. *Set Point* merupakan tekanan awal pipa dalam keadaan tidak ada kebocoran yang dibuka. Dimana nilai *set point* didapatkan dari hasil rata-rata tekanan di titik 1, 2 dan 3. *Set point* titik 1 adalah 13,00 Psi, titik 2 adalah 12,77 Psi dan titik 3 adalah 12,44 Psi. Kemudian dibuat persentase kebocoran yang didapat dari rumus berikut:

$$\%\ \text{Kebocoran} = \frac{Tekanan\ Set\ Point - \text{Tekanan Sekarang}}{Set\ Point}\ X\ 100\%$$

Tabel 4.16 Tekanan Ketika *Stop Valve* Ditutup $\pm 54^0$ Kran Kebocoran 1 Dibuka $\pm 30^0$

No.	Titik 1 (Psi)	Titik 2 (Psi)	Titik 3 (Psi)	% Kebocoran
1	11,2735	10,9952	10,7122	15,48910395
2	11,056	11,1048	10,7557	15,04207169
3	11,056	10,9735	10,7557	14,75713184
4	11,1212	10,9735	10,7993	14,39765389
5	11,0995	11,0995	10,8862	15,69381043
6	11,0995	10,9083	10,8862	13,40490104
7	11,1865	11,0178	10,6688	13,60575215
8	11,0342	10,9952	10,8645	14,49471898
9	11,0125	10,9083	10,9958	15,09252128
10	11,2735	10,9735	10,8862	12,64035584

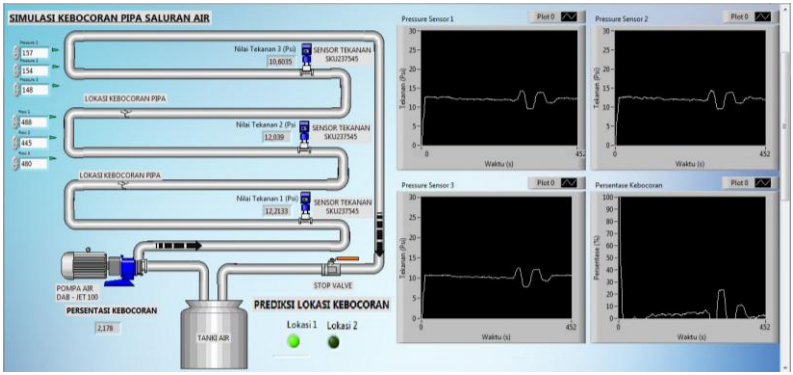
Dari Tabel 4.16 dapat dilihat bahwa pada saat *stop valve* ditutup $\pm 54^0$ dan titik kebocoran 1 dibuka sebesar $\pm 30^0$ dan titik kebocoran

2 ditutup, tekanan yang pada titik 1, 2 dan 3 adalah seperti pada Tabel 4.16.

Tabel 4.17 Tekanan Ketika *Stop Valve* Ditutup ±54^0 Kran Kebocoran 2 Dibuka ±30^0

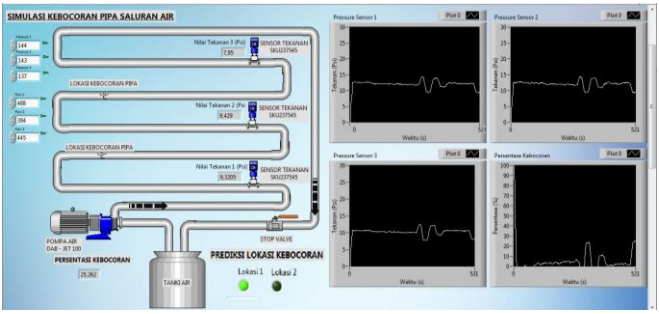
No.	Titik 1 (Psi)	Titik 2 (Psi)	Titik 3 (Psi)	%Kebocoran
1	10,8167	10,669	10,3208	18,91346882
2	10,7297	10,495	10,212	17,54946786
3	10,7732	10,5167	10,3642	16,93754819
4	10,534	10,5385	10,2773	18,91746269
5	10,7732	10,4515	10,212	18,17222024
6	10,5775	10,5167	10,3208	17,47739454
7	10,5122	10,4297	10,473	18,81342585
8	10,882	10,4515	10,1467	15,67413423
9	10,621	10,5385	10,1903	18,11102544
10	10,7297	10,4297	10,4947	16,85432439

Dari Tabel 4.17 dapat dilihat bahwa pada saat *stop valve* ditutup ±54^0 dan titik kebocoran 2 dibuka sebesar ±30^0 dan titik kebocoran 2 ditutup, tekanan yang pada titik 1, 2 dan 3 adalah seperti pada Tabel 4.17.



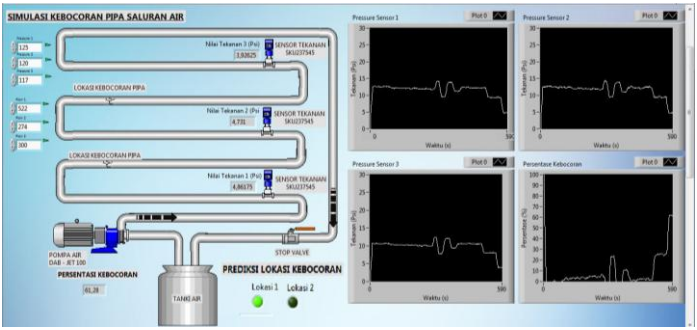
Gambar 4.10. Tampilan *LabVIEW* Alat Simulasi Keadaan Normal

Pada Gambar 4.10 alat simulasi dalam keadaan normal. Semua titik kebocoran ditutup dan *stop valve* ditutup ±54^0. Tekanan di titik 1 terbaca 12,21 Psi. Tekanan pada titik 2 terbaca 12,03 Psi. Tekanan pada titik 3 terbaca 10,60 Psi. Persentase kebocoran yang terbaca adalah 2,17% dan lampu kebocoran lokasi 1 menyala.



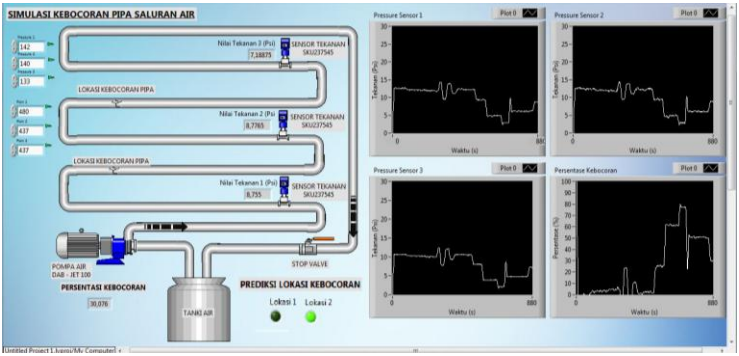
Gambar 4.11. Tampilan *LabVIEW* Alat Simulasi Keadaan Kebocoran 1 Dibuka 20^0

Pada Gambar 4.11 alat simulasi dalam keadaan keadaan Kebocoran 1 dibuka 20^0. Titik kebocoran 2 ditutup dan *stop valve* ditutup ±54^0. Tekanan di titik 1 terbaca 9,32 Psi. Tekanan pada titik 2 terbaca 9,42 Psi. Tekanan pada titik 3 terbaca 7,95 Psi. Persentase kebocoran yang terbaca adalah 25,26% dan lampu kebocoran lokasi 1 menyala.



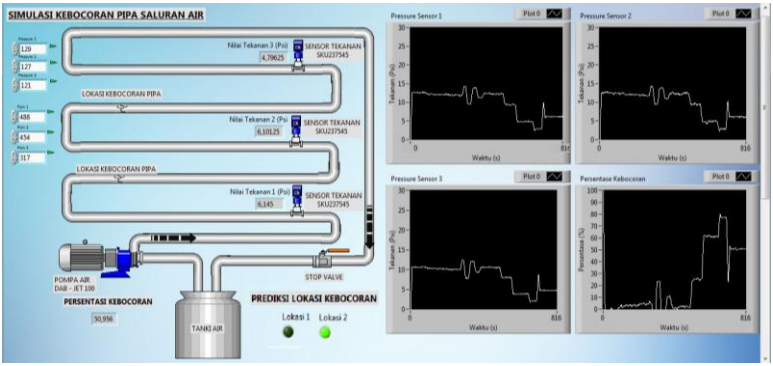
Gambar 4.12. Tampilan *LabVIEW* Alat Simulasi Keadaan Kebocoran 1 Dibuka 30^0

Pada Gambar 4.12 alat simulasi dalam keadaan keadaan Kebocoran 1 dibuka 30^0. Titik kebocoran 2 ditutup dan *stop valve* ditutup $\pm 54^0$. Tekanan di titik 1 terbaca 4,86 Psi. Tekanan pada titik 2 terbaca 4,73 Psi. Tekanan pada titik 3 terbaca 1,92 Psi. Persentase kebocoran yang terbaca adalah 61,28% dan lampu kebocoran lokasi 1 menyala.



Gambar 4.13. Tampilan *LabVIEW* Alat Simulasi Keadaan Kebocoran 2 Dibuka 20^0

Pada Gambar 4.13 alat simulasi dalam keadaan keadaan Kebocoran 2 dibuka 20^0. Titik kebocoran 1 ditutup dan *stop valve* ditutup $\pm 54^0$. Tekanan di titik 1 terbaca 8,72 Psi. Tekanan pada titik 2 terbaca 8,77 Psi. Tekanan pada titik 3 terbaca 7,18 Psi. Dan persentase kebocoran yang terbaca adalah 30,87% dan lampu kebocoran lokasi 2 menyala.



Gambar 4.14. Tampilan *LabVIEW* Alat Simulasi Keadaan Kebocoran 2 Dibuka 30^0

Pada Gambar 4.14 alat simulasi dalam keadaan keadaan Kebocoran 2 dibuka 30^0. Titik kebocoran 1 ditutup dan *stop valve* ditutup $\pm 54^0$. Tekanan di titik 1 terbaca 6,14 Psi. Tekanan pada titik 2 terbaca 6,10 Psi. Tekanan pada titik 3 terbaca 4,79 Psi. Persentase kebocoran yang terbaca adalah 50,93% dan lampu kebocoran lokasi 2 menyala.

4.6 Pengujian Lokasi Kebocoran

Pada pengujian lokasi ini diambil data nyala lampu prediksi lokasi kebocoran *LabVIEW* kemudian dianalisa dimana lokasi kebocoran. Sifat fluida dimana titik yang lokasinya dekat dengan kebocoran akan mengalami *drop* tekanan yang lebih besar dibandingkan dengan titik yang lain. Sehingga rumus yang digunakan untuk memprediksi lokasi kebocoran adalah

$$a = Set\ Point\ 1 - \text{Tekanan } 1$$

$$b = Set\ Point\ 3 - \text{Tekanan } 3$$

Dari persamaan diatas *set point* didapatkan dari rata-rata tekanan awal di titik 1 dan 3. Titik 1 dan 3 dipilih karena kedua titik ini memiliki jarak terdekat dengan kebocoran. Apabila nilai a > b maka drop tekanan paling besar terjadi di lokasi 1 maka kebocoran diprediksi ada di lokasi 1. Apabila a < b maka *drop* tekanan paling besar terjadi di lokasi 2 maka kebocoran diprediksi ada di lokasi 2. Nyala lampu pada *LabVIEW* setelah dimaksukan fungsi tersebut adalah sebagai berikut :

Tabel 4.18 Pengujian Prediksi Lokasi Kebocoran 1 (Kebocoran 1 Dibuka 30^0)

No.	Tampilan *LabVIEW*	Keterangan
Percobaan 1		Lampu kebocoran lokasi 1 menyala
Percobaan 2		Lampu kebocoran lokasi 1 menyala
Percobaan 3		Lampu kebocoran lokasi 1 mati
Percobaan 4		Lampu kebocoran lokasi 1 menyala
Percobaan 5		Lampu kebocoran lokasi 1 menyala

Berdasarkan Tabel 4.18, ketika kebocoran 1 dibuka 30^0, Percobaan 1,2,4 dan 5 lampu berhasil memprediksi lokasi kebocoran dengan tepat. Tetapi pada percobaan 3 lampu gagal memprediksi lokasi kebocoran. Hal ini dikareanakan nilai selisih tekanan yang terbaca terlalu kecil.

Tabel 4.19 Pengujian Prediksi Lokasi Kebocoran 2 (Kebocoran 2 Dibuka 30^0)

No.	Tampilan *LabVIEW*	Keterangan
Percobaan 1		Lampu kebocoran lokasi 2 menyala
Percobaan 2		Lampu kebocoran lokasi 2 mati
Percobaan 3		Lampu kebocoran lokasi 2 menyala
Percobaan 4		Lampu kebocoran lokasi 2 mati
Percobaan 5		Lampu kebocoran lokasi 2 menyala

Berdasarkan Tabel 4.19, ketika kebocoran 2 dibuka 30^0, Percobaan 1,3 dan 5 lampu berhasil memprediksi lokasi kebocoran dengan tepat. Tetapi pada percobaan 2 dan 4 lampu gagal memprediksi lokasi kebocoran. Hal ini dikareanakan nilai selisih tekanan yang terbaca terlalu kecil. Dari Tabel 4.18 dan 4.19 keakuratan prediksi lokasi kebocoran adalah 70%.

BAB V
PENUTUP

Setelah melakukan perancangan dan pembuatan alat serta pengujian dan analisa, maka dapat ditarik kesimpulan dan saran dari kegiatan yang telah dilakukan untuk pengembangan Tugas Akhir ini.

5.1 Kesimpulan

Dari seluruh tahapan yang sudah dilaksanakan pada penyusunan Tugas Akhir ini, mulai dari studi literatur, perancangan dan pembuatan sampai pada pengujiannya maka dapat disimpulkan bahwa :

1. Ketika terjadi kebocoran pipa, tekanan dalam pipa akan mengalami penurunan tergantung pada besar kebocoran.
2. Pada keadaan tidak bocor presentase kebocoran sudah menunjukan angka ±3%
3. Nilai kebocoran yang dapat dideteksi dari simulasi ini yaitu sekitar ±10% dari jumlah tekanan total dari air.
4. Nilai *error* dari sensor paling besar berkisar ±8,67% ketika kalibrasi.
5. Keakuratan prediksi lokasi kebocoran sebesar 70%.
6. Nilai tekanan pada masing masing titik mengalami penurunan karena panjang pipa. Penurunan tekanan berkisar ± 0,5 Psi tiap 1 m.
7. Perbedaan tekanan paling kecil yang dapat dideteksi oleh sensor adalah ±0,5 Psi

5.2 Saran

Untuk lebih memperbaiki dan menyempurnakan kinerja dari alat ini, maka perlu disarankan :

1. Digunakan pipa besi yang mampu beroperasi pada tekanan tinggi (diatas 30 Psi) dan menggunakan pompa yang lebih besar agar dapat menghasilkan tekanan yang lebih tinggi.
2. Penambahan RTC dan SD *Card* agar dapat menyimpan data
3. Penambahan SIM 900 agar dapat SMS otomatis.

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

DAFTAR PUSTAKA

[1] Wafieyuddin, *DMA 214 Klampis Wisma Mukti*, ***Jurnal***, PDAM Surya Sembada Kota Surabaya, Surabaya, 2018

[2] Halliday, David, ***Fisika Edisi ke 3 Jilid 1***, Erlangga, Jakarta, 1987

[3] Giancolli, Douglas, ***Fisika Edisi Kelima Jilid 1***, Erlangga, Jakarta, 2001

[4] Fahruzi, Akhmad, "*Rancang Bangun Sensor Level Berbasis Sensor Tekanan Pada Tangki Proses Minyak Kelapa*", ***Tugas Akhir***. Program Magister Bidang Keahlian Fisika Instrumentasi, Jurusan Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember, Surabaya. 2015

[5] Vatmawati, Ninik, "*Pembuatan Pendeteksi Gas Beracun Portable Untuk Penggali Sumur Berbasis Mikrokontroler*", ***Tugas Akhir***, Jurusan D3 Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya. 2016

[6] Fadilah, Farrah, "*Telemetring Kebocoran Pipa Pada Distribusi Air Dengan Komunikasi Ethernet*", ***Tugas Akhir.*** Jurusan D3 Teknik Elektro. Fakultas Vokasi, Institut Teknologi Sepuluh Nopember, Surabaya. 2018

[7] Maulana, Restu, "*Rancang Bangun Kulkas Portable Menggunakan Termoelektrik Berbasis Mikrokontroler". **Tugas Akhir***. Jurusan D3 Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya. 2016

[8] Hariyanto, Duwi, "*Deteksi Letak Kebocoran Pipa Berdasarkan Analisis debit Air Menggunakan teknologi sensor flowmeter berbasis TCP/IP*", ***Skripsi***, Jurusan Fisika, FMIPA, Universitas Lampung, Bandar Lampung. 2016

[9] Susanto, Deki, "*Analisa Distribusi Air pada Pipa Jaringan Distribusi di Sub-Zone Sodakan PDAM Kota Surakarta*", ***Skripsi***, Jurusan Teknik Industri, Fakultas Teknik, Universitas Sebelas Maret, Surakarta. 2007

LAMPIRAN A
PROGRAM ARDUINO

```
#include <LiquidCrystal.h>

#include <SPI.h>

#include <Ethernet.h>

#include <Modbus.h>

#include <ModbusIP.h>


const int sencalc = 100;

const int water = 150;


double press_psi;

long ts;

LiquidCrystal lcd(9, 8, 7, 6, 5, 3);

volatile int NbTopsFan;

int Calc;

int jumlahperulanganbaca=0;

double tekanantotal=0;

double debittotal=0;


ModbusIP mb;
```

```
void rpm ()
{
  NbTopsFan++;
}

void setup()
{
    lcd.begin(16, 2);
    byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
    byte ip[] = { 192, 168, 1, 196 };
    mb.config(mac, ip);
    Serial.begin(9600);
    attachInterrupt(0, rpm, RISING);
    mb.addIreg(sencalc);
    mb.addIreg(water);
    sei();
}

void loop ()
{
```

```
mb.task();
if (millis() > ts + 1000) {
    ts = millis();
    cli();

    int val=analogRead(A0);
    press_psi = (((val*0.015)+1.06)*14.5-37.3);
    int sensorVal = val;
    mb.Ireg(sencalc, sensorVal);

    Calc = (NbTopsFan *60 / 7);
    mb.Ireg(water, Calc);

    NbTopsFan = 0;

    jumlahperulanganbaca=jumlahperulanganbaca+1;
    tekanantotal=tekanantotal+press_psi;
    debittotal=debittotal+Calc;
    if (jumlahperulanganbaca==10){
      Serial.print(tekanantotal/10);
      Serial.println (" Psi");
```

```
        Serial.print (Calc);
        Serial.println (" L/hours");

        lcd.setCursor(0,0);
        lcd.print("Psi : ");
        lcd.print(tekanantotal/10);

        lcd.setCursor(0,1);
        lcd.print("Flow : ");
        lcd.print(Calc);
        lcd.print(" L/hours");

        jumlahperulanganbaca=0;
        tekanantotal=0;
         }

      delay(100);
      sei();
  }
}
```

LAMPIRAN B
DATASHEET

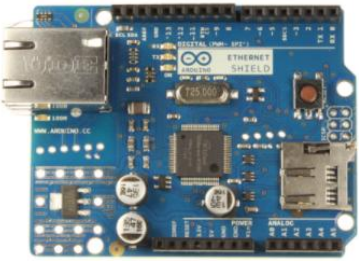
B.1 *Pressure Sensor* SKU237545

Working Voltage	*DC 5.0V*
Output Voltage	*DC 0.5-4.5 V*
Sensor material	*Carbon steel alloy*
Working Current	*≤10 mA*
Working Pressure Range	*0-1.2 MPa*
The Biggest Pressure	*2.4 MPa*
Cable length	*19cm*
Destroy Pressure	*3.0 MPa*
Working TEMP. Range	*0-85℃*
Storage Temperature Range	*0-100℃*
Measuring Error	*±1.5 %FSO*
Temperature Range Error:	*±3.5 %FSO*
Response Time	*≤2.0 ms*
Cycle Life	*500,000 pcs*
Application	*non-corrosive gas liquid measurement*

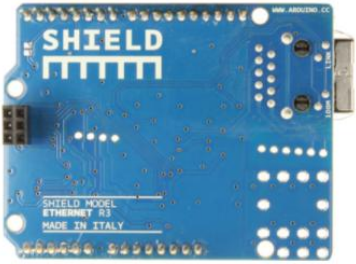
Features:
Easy removal, carbon steel connection more firmly.
Stainless steel is durable, sealed waterproof line, imported chips.
Wiring: red +, black -, yellow output

B.2 Arduino *Ethernet Shield* V1

Arduino Ethernet Shield V1



Arduino Ethernet Shield R3 Front



Arduino Ethernet Shield R3 Back

Overview

NOTE: this product is currently retired and the documentation will not be kept up-to-date

The Arduino Ethernet Shield V1 connects your Arduino to the internet in mere minutes. Just plug this module onto your Arduino board, connect it to your network with an RJ45 cable (not included) and follow a few simple instructions to start controlling your world through the internet. As always with Arduino, every element of the platform – hardware, software and documentation – is freely available and open-source. This means you can learn exactly how it's made and use its design as the starting point for your own circuits. Hundreds of thousands of Arduino boards are already fueling people's creativity all over the world, everyday. Join us now, Arduino is you!

- Requires an Arduino board (not included)
- Operating voltage 5V (supplied from the Arduino Board)
- Ethernet Controller: W5100 with internal 16K buffer
- Connection speed: 10/100Mb
- Connection with Arduino on SPI port

Description

The Arduino Ethernet Shield V1 allows an Arduino board to connect to the internet. It is based on the Wiznet W5100 ethernet chip (datasheet). The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the Ethernet library to write sketches which connect to the internet using the shield. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

The most recent revision of the board exposes the 1.0 pinout on rev 3 of the Arduino UNO board.

The Ethernet Shield V1 has a standard RJ-45 connection, with an integrated line transformer and Power over Ethernet enabled.

There is an onboard micro-SD card slot, which can be used to store files for serving over the network. It is compatible with all the Arduino/Genuino boards. The on-board micro SD card reader is accessible through the SD Library. When working with this library, SS is on Pin 4. The original revision of the shield contained a full-size SD card slot; this is not supported.

The shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up.

Download: arduino-ethernet-shield-06-schematic.pdf, arduino-ethernet-shield-06-reference-design.zip

The current shield has a Power over Ethernet (PoE) module designed to extract power from a conventional twisted pair Category 5 Ethernet cable:

- IEEE802.3af compliant
- Low output ripple and noise (100mVpp)
- Input voltage range 36V to 57V
- Overload and short-circuit protection
- 9V Output
- High efficiency DC/DC converter: typ 75% @ 50% load
- 1500V isolation (input to output)

NB: the Power over Ethernet module is proprietary hardware not made by Arduino, it is a third party accessory. For more information, see the datasheet

The shield does not come with the PoE module built in, it is a separate component that must be added on.

Arduino communicates with both the W5100 and SD card using the SPI bus (through the ICSP header). This is on digital pins 10, 11, 12, and 13 on the Uno and pins 50, 51, and 52 on the Mega. On both boards, pin 10 is used to select the W5100 and pin 4 for the SD card. These pins cannot be used for general I/O. On the Mega, the hardware SS pin, 53, is not used to select either the W5100 or the SD card, but it must be kept as an output or the SPI interface won't work.

Note that because the W5100 and SD card share the SPI bus, only one can be active at a time. If you are using both peripherals in your program, this should be taken care of by the corresponding libraries. If you're not using one of the peripherals in your program, however, you'll need to explicitly deselect it. To do this with the SD card, set pin 4 as an output and write a high to it. For the W5100, set digital pin 10 as a high output.

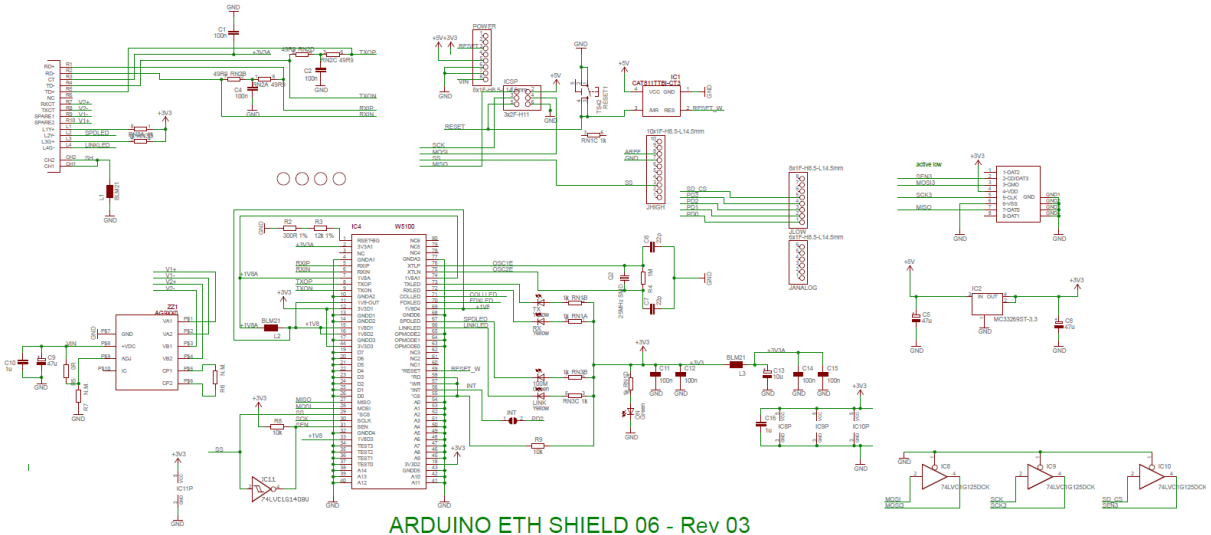
The shield provides a standard RJ45 ethernet jack.

The reset button on the shield resets both the W5100 and the Arduino board.

The shield contains a number of informational LEDs:

- PWR: indicates that the board and shield are powered
- LINK: indicates the presence of a network link and flashes when the shield transmits or receives data
- FULLD: indicates that the network connection is full duplex
- 100M: indicates the presence of a 100 Mb/s network connection (as opposed to 10 Mb/s)
- RX: flashes when the shield receives data
- TX: flashes when the shield sends data
- COLL: flashes when network collisions are detected

The solder jumper marked "INT" can be connected to allow the Arduino board to receive interrupt-driven notification of events from the W5100, but this is not supported by the Ethernet library. The jumper connects the INT pin of the W5100 to digital pin 2 of the Arduino.



ARDUINO ETH SHIELD 06 - Rev 03

Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE
Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.
The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

ARDUINO is a registered trademark.

B.2.1 Wiznet W5100

WIZnet

W5100 Datasheet

The W5100 is a full-featured, single-chip Internet-enabled 10/100 Ethernet controller designed for embedded applications where ease of integration, stability, performance, area and system cost control are required. The W5100 has been designed to facilitate easy implementation of Internet connectivity without OS. The W5100 is IEEE 802.3 10BASE-T and 802.3u 100BASE-TX compliant.

The W5100 includes fully hardwired, market-proven TCP/IP stack and integrated Ethernet MAC & PHY. Hardwired TCP/IP stack supports TCP, UDP, IPv4, ICMP, ARP, IGMP and PPPoE which has been proven in various applications for several years. 16Kbytes internal buffer is included for data transmission. No need of consideration for handling Ethernet Controller, but simple socket programming is required.

For easy integration, three different interfaces like memory access way, called direct, indirect bus and SPI, are supported on the MCU side.

Target Applications

The W5100 is well suited for many embedded applications, including:

- Home Network Devices: Set-Top Boxes, PVRs, Digital Media Adapters
- Serial-to-Ethernet: Access Controls, LED displays, Wireless AP relays, etc.
- Parallel-to-Ethernet: POS / Mini Printers, Copiers
- USB-to-Ethernet: Storage Devices, Network Printers
- GPIO-to-Ethernet: Home Network Sensors
- Security Systems: DVRs, Network Cameras, Kiosks
- Factory and Building Automations
- Medical Monitoring Equipments
- Embedded Servers

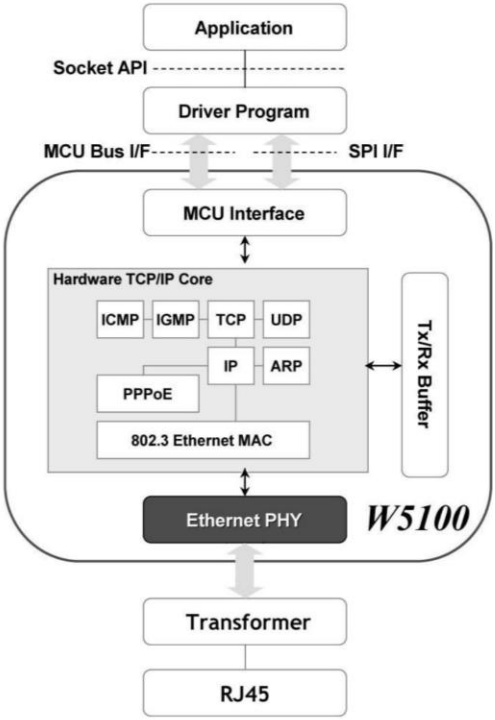
© Copyright 2008 WIZnet Co., Inc. All rights reserved.

Features

- Support Hardwired TCP/IP Protocols : TCP, UDP, ICMP, IPv4 ARP, IGMP, PPPoE, Ethernet
- 10BaseT/100BaseTX Ethernet PHY embedded
- Support Auto Negotiation (Full-duplex and half duplex)
- Support Auto MDI/MDIX
- Support ADSL connection (with support PPPoE Protocol with PAP/CHAP Authentication mode)
- Supports 4 independent sockets simultaneously
- Not support IP Fragmentation
- Internal 16Kbytes Memory for Tx/Rx Buffers
- 0.18 μm CMOS technology
- 3.3V operation with 5V I/O signal tolerance
- Small 80 Pin LQFP Package
- Lead-Free Package
- Support Serial Peripheral Interface(SPI MODE 0, 3)
- Multi-function LED outputs (TX, RX, Full/Half duplex, Collision, Link, Speed)

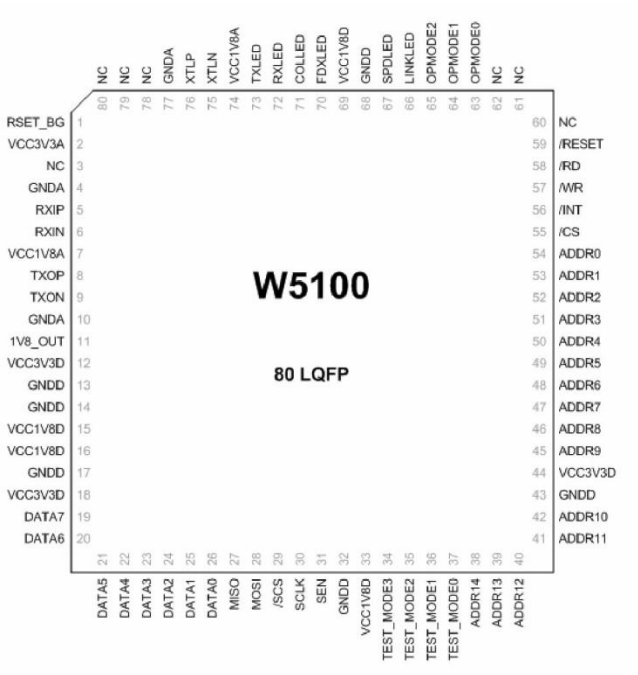
© Copyright 2008 WIZnet Co., Inc. All rights reserved.

Block Diagram



© Copyright 2008 WIZnet Co., Inc. All rights reserved.

1. Pin Assignment



Pinout W5100

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

7. Electrical Specifications

Absolute Maximum Ratings

Symbol	Parameter	Rating	Unit
V_{DD}	DC Supply voltage	-0.5 to 3.6	V
V_{IN}	DC input voltage	-0.5 to 5.5 (5V tolerant)	V
I_{IN}	DC input current	±5	mA
T_{OP}	Operating temperature	-40 to 85	°C
T_{STG}	Storage temperature	-55 to 125	°C

***COMMENT**: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage.

DC Characteristics

Symbol	Parameter	Test Condition	Min	Typ	Max	Unit
V_{DD}	DC Supply voltage	Junction temperature is from -55°C to 125°C	3.0		3.6	V
V_{IH}	High level input voltage		2.0		5.5	V
V_{IL}	Low level input voltage		- 0.5		0.8	V
V_{OH}	High level output voltage	I_{OH} = 2, 4, 8, 12, 16, 24 mA	2.0		3.6	V
V_{OL}	Low level output voltage	I_{OL} = -2, -4, -8, -12, -16, -24 mA	0.0		0.4	V
I_I	Input Current	$V_{IN} = V_{DD}$			±5	μA

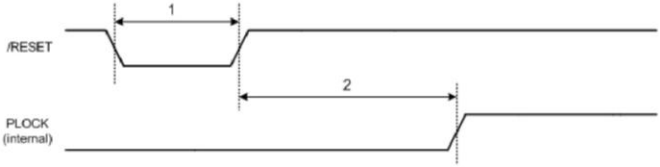
POWER DISSIPATION

Symbol	Parameter	Test Condition	Min	Typ	Max	Unit
P_{10Base}	Power consumption in 10BaseT	Vcc 3.3V Temperature 25°C	-	138	183	mA
$P_{100Base}$	Power consumption in 100BaseT	Vcc 3.3V Temperature 25°C	-	146	183	mA

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

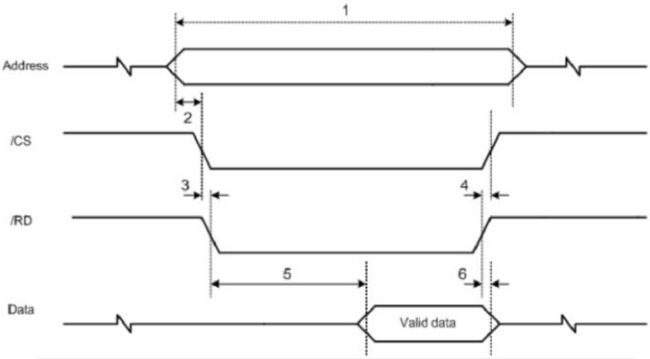
AC Characteristics

Reset Timing



	Description	Min	Max
1	Reset Cycle Time	2 us	-
2	/RESET to internal PLOCK	-	10 ms

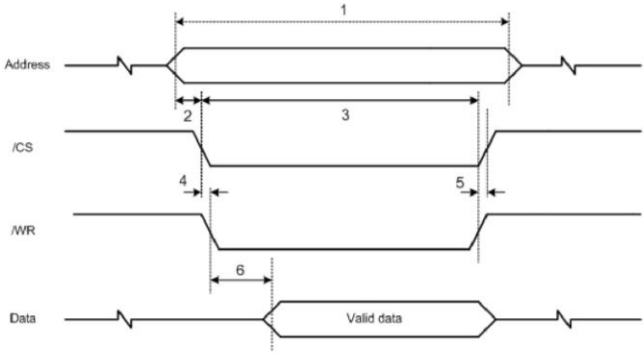
Register/Memory READ Timing



	Description	Min	Max
1	Read Cycle Time	80 ns	-
2	Valid Address to /CS low time	8 ns	-
3	/CS low to /RD low time	-	1 ns
4	/RD high to /CS high time	-	1 ns
5	/RD low to Valid Data Output time	-	80 ns
6	/RD high to Data High-Z Output time	-	1 ns

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

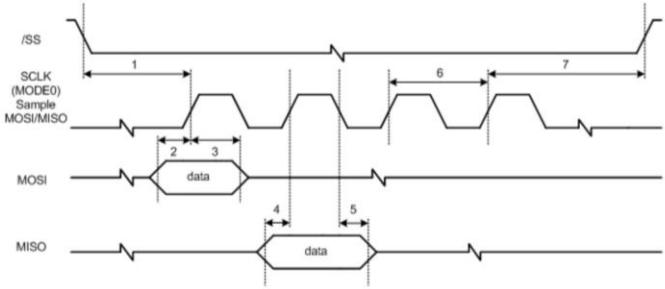
Register/Memory WRITE Timing



	Description	Min	Max
1	Write Cycle Time	70 ns	-
2	Valid Address to /CS low time	7 ns	-
3	/CS low to /WR high time	70 ns	-
4	/CS low to /WR low time	-	1 ns
5	/WR high to /CS high time	-	1 ns
6	/WR low to Valid Data time	-	14 ns

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

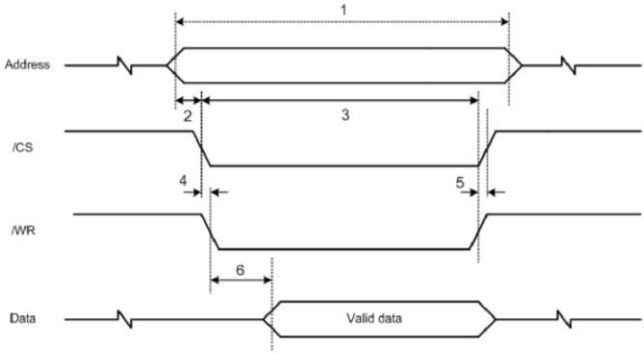
SPI Timing



	Description	Mode	Min	Max
1	/SS low to SCLK high	Slave	21 ns	-
2	Input setup time	Slave	7 ns	-
3	Input hold time	Slave	28 ns	-
4	Output setup time	Slave	7 ns	14 ns
5	Output hold time	Slave	21 ns	-
6	SCLK time	Slave	70 ns	
7	SCLK high to /SS high	Slave	21ns	

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

Register/Memory WRITE Timing



	Description	Min	Max
1	Write Cycle Time	70 ns	-
2	Valid Address to /CS low time	7 ns	-
3	/CS low to /WR high time	70 ns	-
4	/CS low to /WR low time	-	1 ns
5	/WR high to /CS high time	-	1 ns
6	/WR low to Valid Data time	-	14 ns

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

Crystal Characteristics

Parameter	Range
Frequency	25 MHz
Frequency Tolerance (at 25℃)	±30 ppm
Shunt Capacitance	7pF Max
Drive Level	100uW
Load Capacitance	27pF
Aging (at 25℃)	±3ppm / year Max

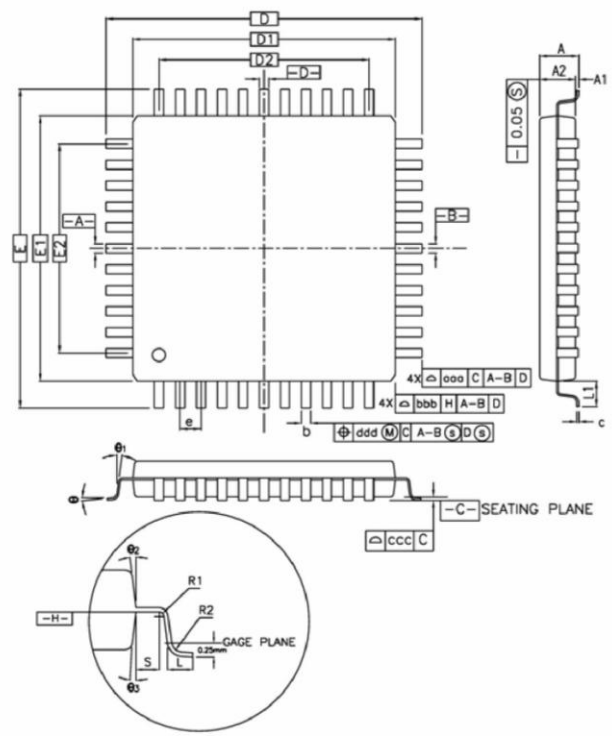
Transformer Characteristics

Parameter	Transmit End	Receive End
Turn Ratio	1:1	1:1
Inductance	350 uH	350 uH

Symmetrical TX & RX channels for auto MDI/MDIX capability

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

Package Descriptions



☞ Above diagram shows PIN dimension. All 80 pins are not displayed.

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	-	-	1.60	-	-	0.063
A1	0.05	-	0.15	0.002	-	0.006

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

A2	1.35	1.40	1.45	0.053	0.055	0.057
D		12.00 BSC.			0.472 BSC.	
D1		10.00 BSC.			0.393 BSC.	
E		12.00 BSC.			0.472 BSC.	
E1		10.00 BSC.			0.393 BSC.	
R2	0.08	-	0.20	0.003	-	0.008
R1	0.08	-	-	0.003	-	-
θ	0°	3.5°	7°	0°	3.5°	7°
θ_1	0°	-	-	0°	-	-
θ_2	11°	12°	13°	11°	12°	13°
θ_3	11°	12°	13°	11°	12°	13°
c	0.09	-	0.20	0.004	-	0.008
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00 REF			0.039 REF	
S	0.20	-	-	0.008	-	-
b	0.13	0.16	0.23	0.005	0.006	0.009
e		0.40 BSC			0.016 BSC	
D2		7.60			0.299	
E2		7.60			0.299	
aaa		0.20			0.008	
bbb		0.20			0.008	
ccc		0.08			0.003	
ddd		0.07			0.003	

Note :

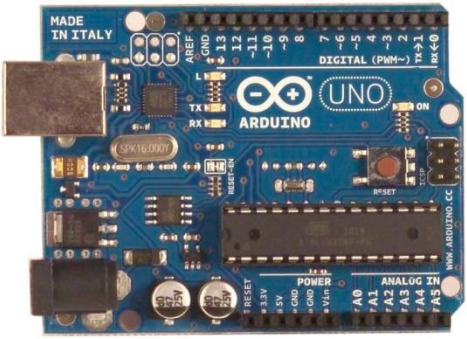
1. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
ALLOWABLE PROTRUSION IS 0.25mm PER SIDE. D1 AND E1 ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD MISMATCH.

2. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.
ALLOWABLE DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED THE MAXIMUM b DIMENSION BY MORE THAN 0.08mm.
DAMBAR CAN NOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT. MINIMUM SPACE BETWEEN PROTRUSION AND AN ADJACENT LEAD IS 0.07mm FOR 0.4mm AND 0.5mm PITCH PACKAGES.

© Copyright 2008 WIZnet Co., Inc. All rights reserved.

B.3 Arduino UNO

Arduino UNO



Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduno, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Index

Technical Specifications — Page 2

How to use Arduino
Programming Enviroment, Basic Tutorials — Page 6

Terms & Conditions — Page 7

Enviromental Policies
half sqm of green via Impatto Zero® — Page 7

radiospares RADIONICS ALLIED ELECTRONICS

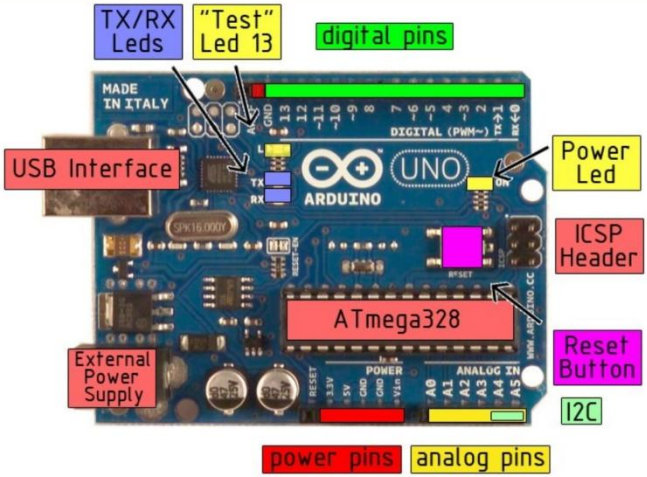
Technical Specification

EAGLE files: arduino-duemilanove-uno-design.zip Schematic: arduino-uno-schematic.pdf

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. TThese pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

radiospares RADIONICS ALLIED ELECTRONICS

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **I^2C: 4 (SDA) and 5 (SCL).** Support I^2C (TWI) communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and Atmega328 ports.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).

radiospares RADIONICS

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

radiospares RADIONICS

How to use Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platoform program. You'll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. *http://arduino.cc/en/Guide/HomePage*

Linux Install **Windows Install** **Mac Install**

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

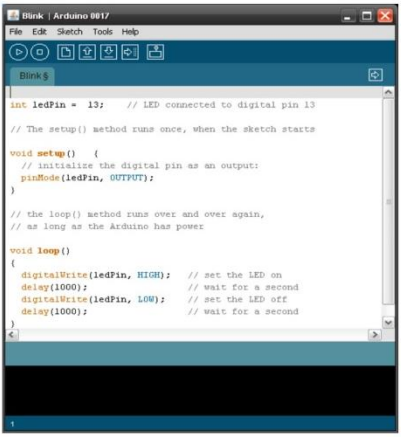
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink

Once you have your skecth you'll see something very close to the screenshot on the right.

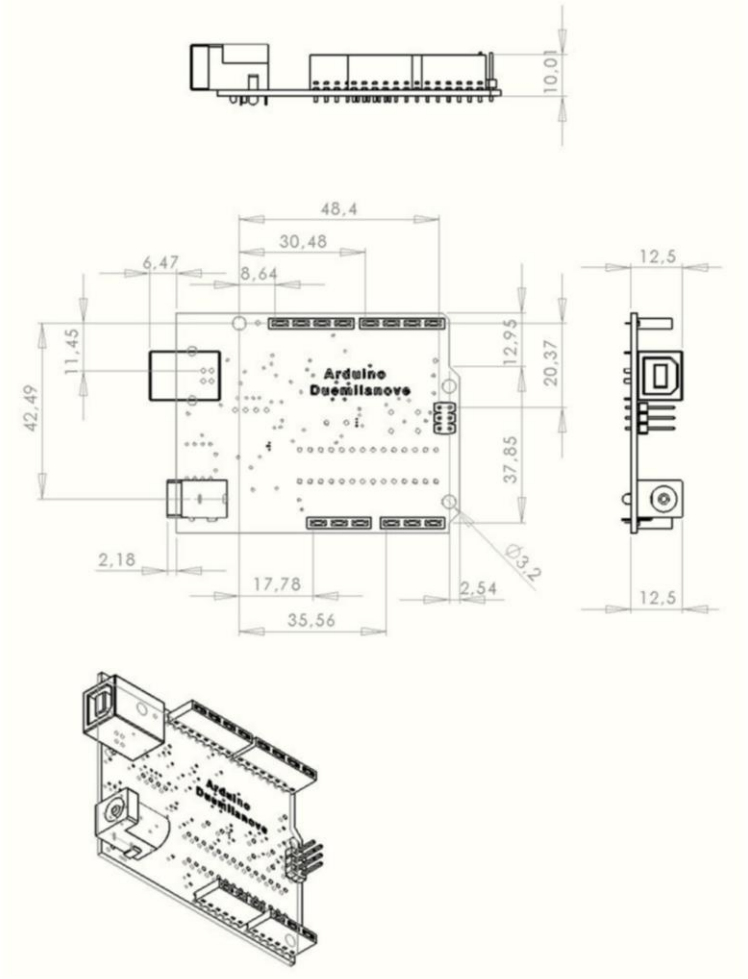
In **Tools>Board** select

Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.



radiospares RADIONICS

Dimensioned Drawing



radiospares RADIONICS

Terms & Conditions

1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

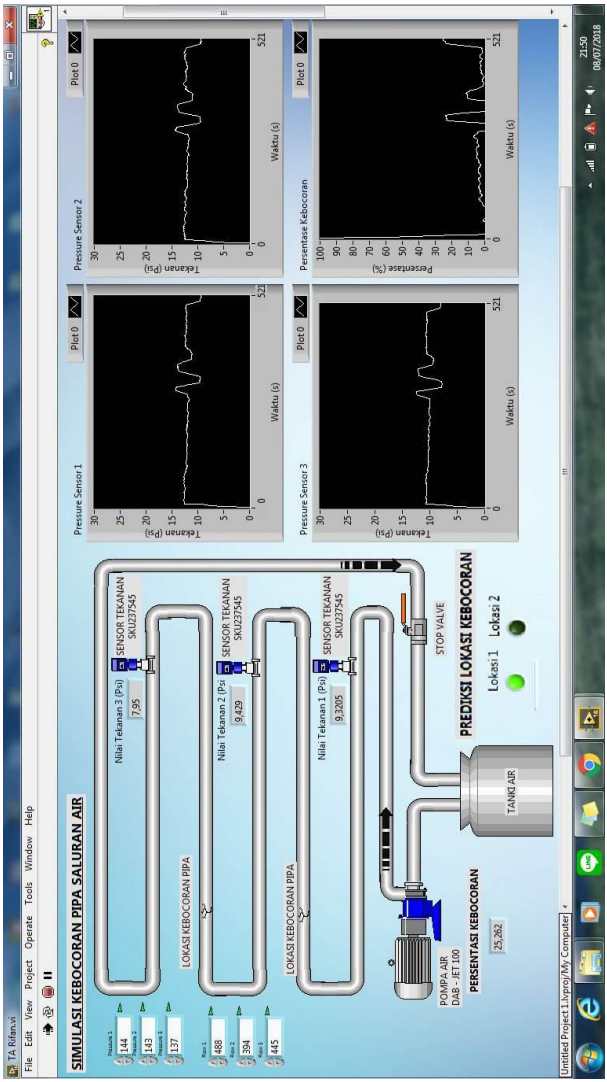
Enviromental Policies

The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.

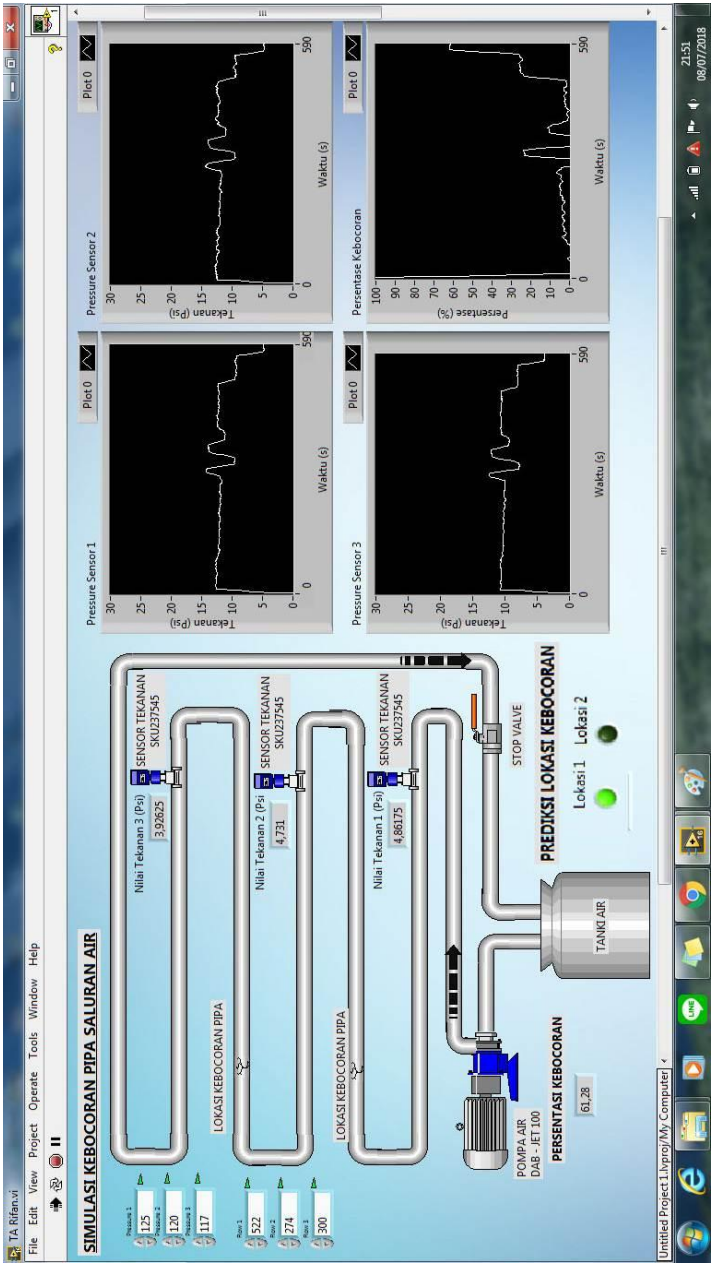
-----Halaman ini sengaja dikosongkan-----

LAMPIRAN C
TAMPILAN *LABVIEW*

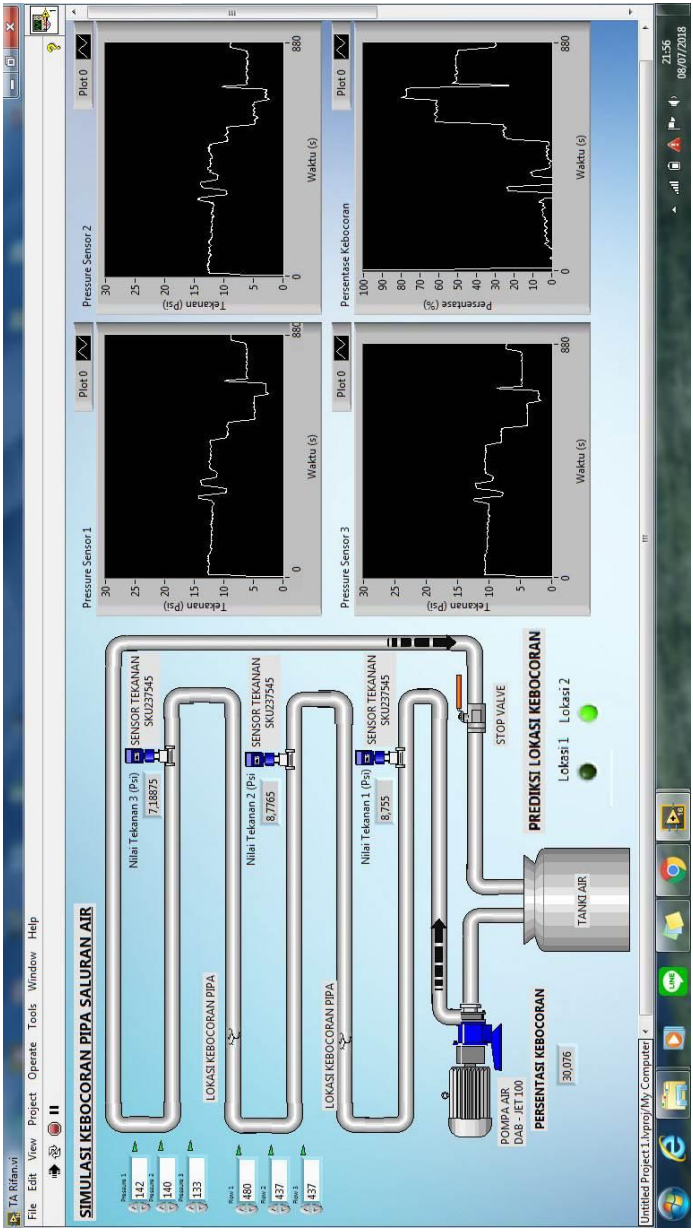
Gambar LabVIEW ketika Kebocoran 1 dibuka 20^0



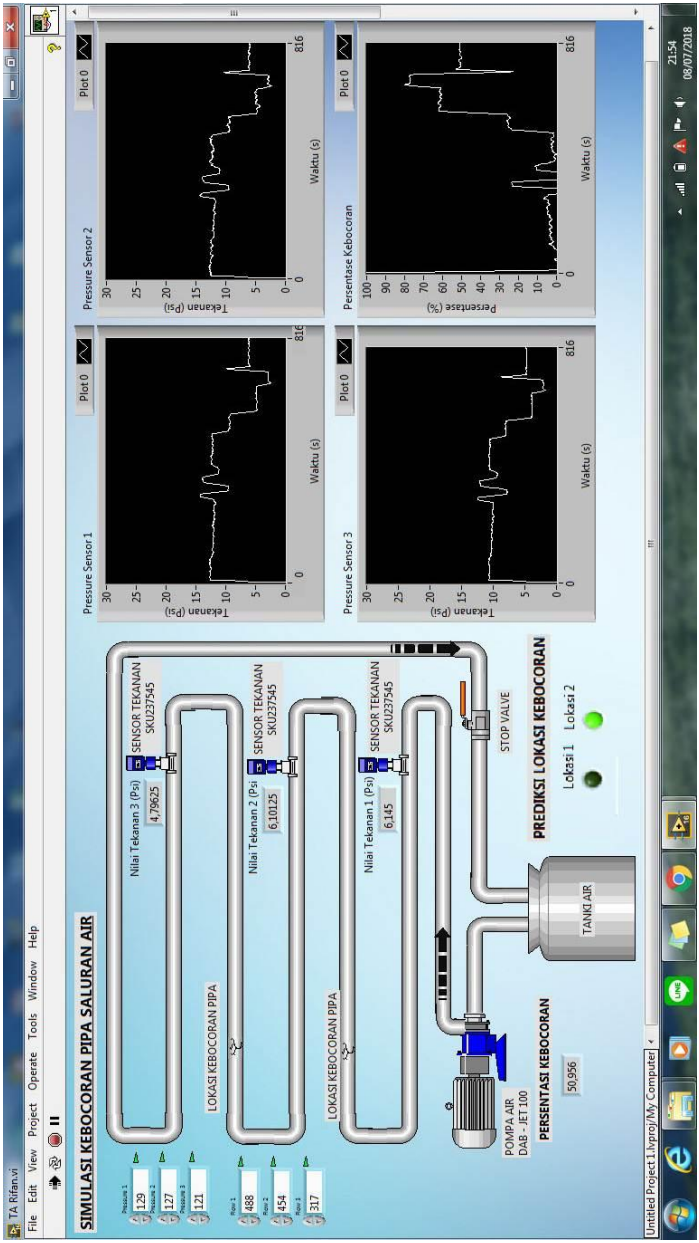
Gambar LabVIEW ketika Kebocoran 1 dibuka 30^0



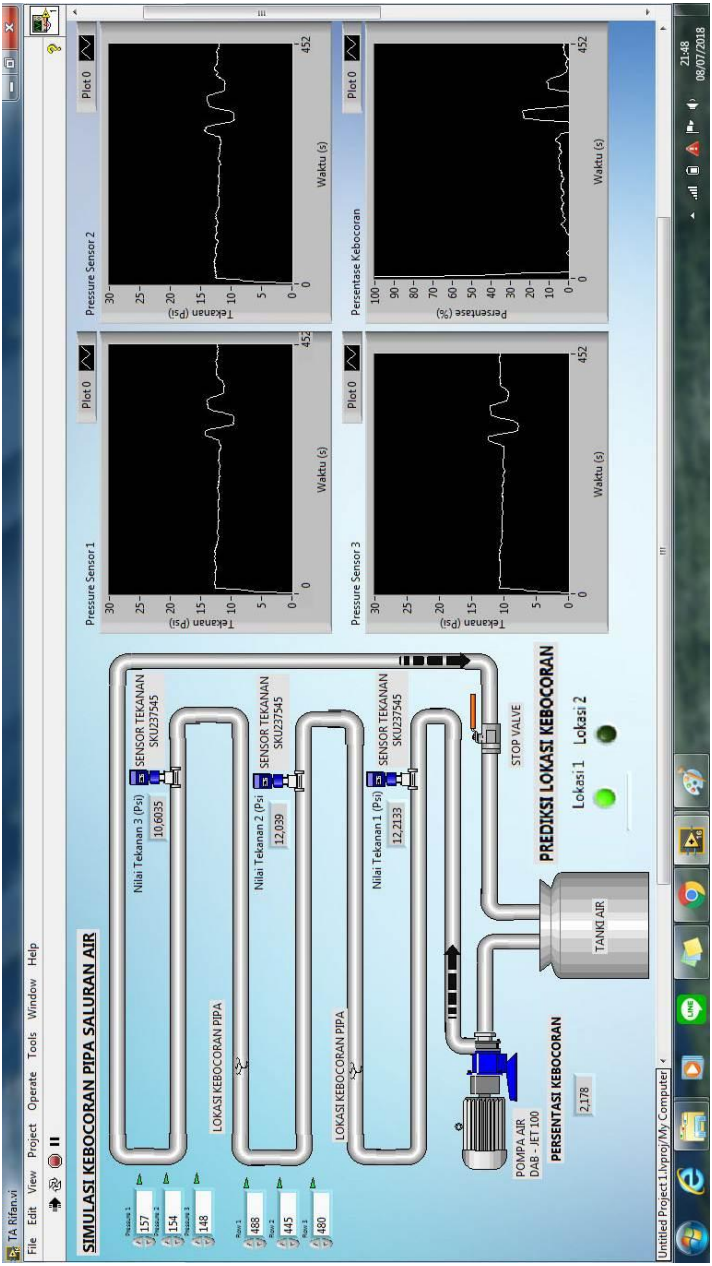
Gambar LabVIEW ketika Kebocoran 2 dibuka 20^0



Gambar LabVIEW ketika Kebocoran 2 dibuka 30^0



Gambar LabVIEW ketika Kebocoran 1 dan 2 Ditutup



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

LAMPIRAN D
GAMBAR ALAT

Gambar *Prototype* Tampak Samping

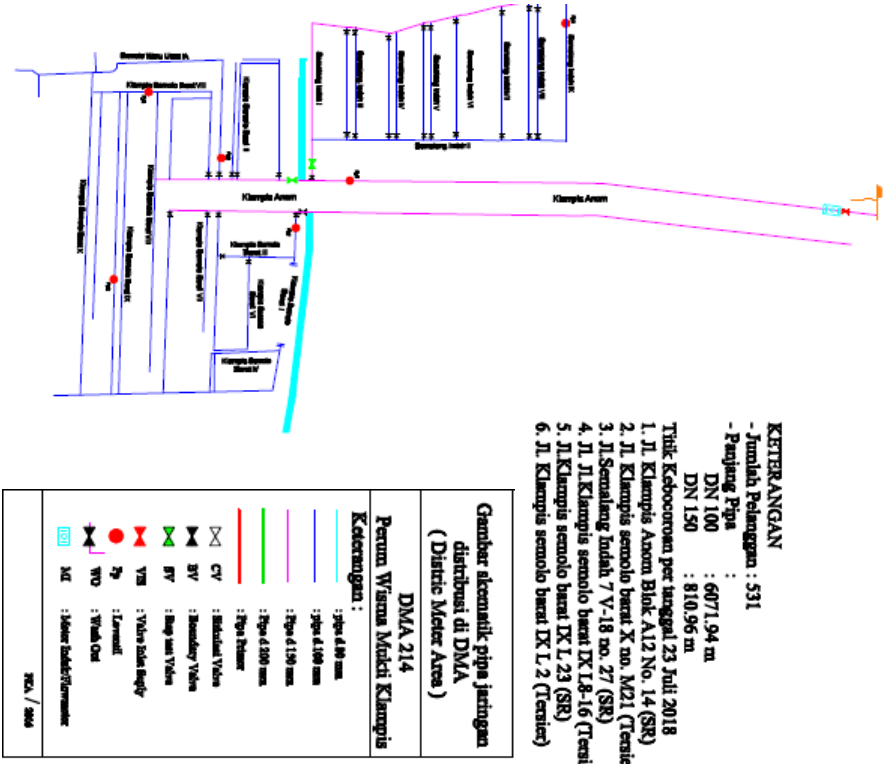
Gambar *Prototype* Tampak Depan

Dokumentasi Kran Kebocoran Saat Dibuka ±20°

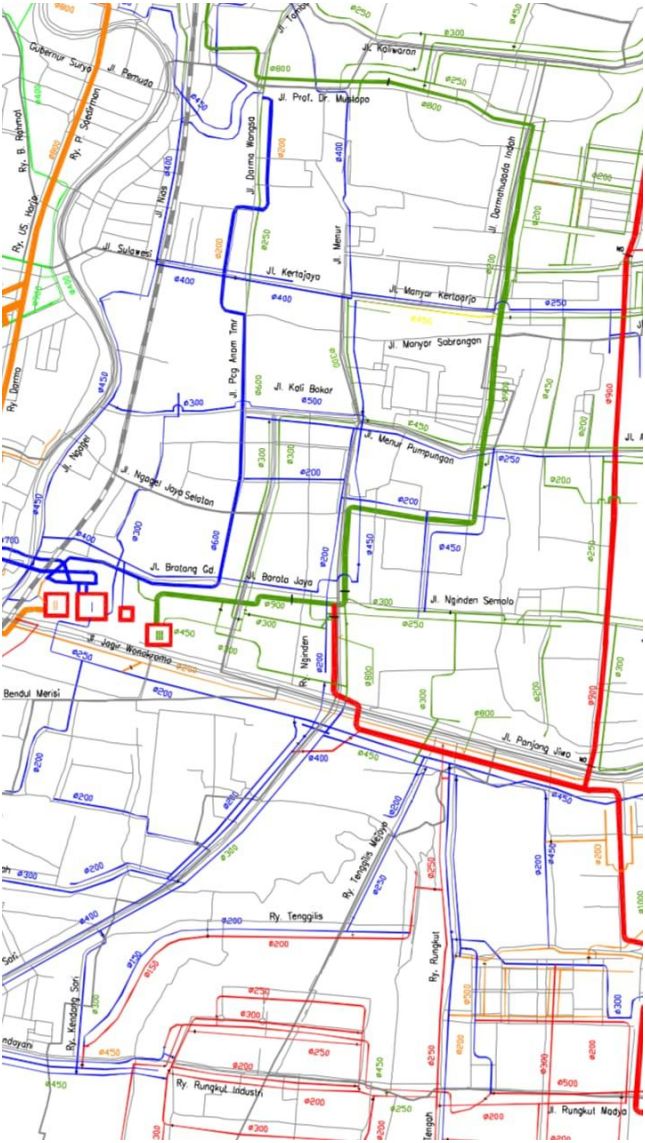
Dokumentasi Kran Kebocoran Saat Dibuka ±30°

LAMPIRAN E
DATA PDAM SURABAYA

Gambar Piping Diagram Kebocoran Daerah Klampis



Gambar Distribusi Air di Surabaya



DAFTAR RIWAYAT HIDUP

Nama : Rifan Syah Rijal
TTL : Mojokerto, 26 Juli 1996
Jenis Kelamin : Perempuan
Agama : Islam
Alamat : Jl. Raya Brangkal No. 676, Sooko, Mojokerto
Telp/Hp : 082141452346
E-mail : rifansyahrijal@gmail.com

RIWAYAT PENDIDIKAN

1. 2003 – 2009 : SD Negeri Kedung Maling II
2. 2009 – 2012 : SMP Negeri 1 Sooko
3. 2012 – 2015 : SMA Negeri 1 Sooko
4. 2015 – 2018 : D3 Teknik Elektro, Departemen Teknik Elektro Otomasi – Fakultas Vokasi, Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT. Sun Paper
2. Kerja Praktek di PT. Semen Indonesia

PENGALAMAN ORGANISASI

1. Staff Syiar Lembaga Dakwah Jurusan Salman Al-Farisi (2016-2017), D3 Teknik Elektro, Institut Teknologi Sepuluh Nopember
2. Kepala Departemen Syiar Lembaga Dakwah Jurusan Salman Al-Farisi (2017-2018), D3 Teknik Elektro, Institut Teknologi Sepuluh Nopember