



TUGAS AKHIR – TF 095565

**PERANCANGAN DAN PEMBUATAN HUMAN
MACHINE INTERFACE PADA SIMULATOR
PENGENDALIAN FLOW DAN LEVEL**

**Nafi' Kurniawan
NRP 105 115 00000 064**

**Dosen Pembimbing I
Dr. Ir. Totok Soehartanto, DEA
NIP. 19650309 199002 1 001
Dosen Pembimbing II
Sefi Novendra Patrialova, S.Si., M.T.
NPP. 1991201712053**

**PROGRAM STUDI DIII TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018**



TUGAS AKHIR – TF 095565

PERANCANGAN DAN PEMBUATAN *HUMAN MACHINE INTERFACE* PADA SIMULATOR PENGENDALIAN FLOW DAN LEVEL

**Nafi' Kurniawan
NRP 105 115 00000 064**

**Dosen Pembimbing I
Dr. Ir. Totok Soehartanto, DEA
NIP. 19650309 199002 1 001**

**Dosen Pembimbing II
Sefi Novendra Patrialova, S.Si., M.T.
NPP. 1991201712053**

**PROGRAM STUDI DIII TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018**



FINAL PROJECT - TF 095565

***DESIGN OF HUMAN MACHINE INTERFACE IN FLOW
AND LEVEL PLANT SIMULATOR***

Nafi' Kurniawan
NRP 105 115 00000 063

Supervisor I
Dr. Ir. Totok Soehartanto, DEA
NIP. 19650309 199002 1 001

Supervisor II
Sefi Novendra Patrialova, S.Si., M.T.
NPP. 1991201712053

***DIII INSTRUMENTATION TECHNOLOGY
DEPARTMENT OF INSTRUMENTATION ENGINEERING
FACULTY OF VOCATION
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018***

**PERANCANGAN DAN PEMBUATAN HUMAN MACHINE
INTERFACE PADA SIMULATOR PENGENDALIAN
FLOW DAN LEVEL**

TUGAS AKHIR

Oleh :

**Nafi' Kurniawan
NRP. 105 115 00000 063**

**Surabaya, 20 Juni 2018
Mengetahui dan Menyetujui,**

Dosen Pembimbing I,



**Dr. Ir. Totok Soehartanto, DEA
NIP. 19650309 199002 1 001**

Dosen Pembimbing II,



**Sefi Novendra Patrialova, S.Si., MT.
NIP. 1991201712053**

Ketua Departemen Teknik Instrumentasi,



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

**PERANCANGAN DAN PEMBUATAN *HUMAN MACHINE*
INTERFACE PADA SIMULATOR PENGENDALIAN
FLOW DAN LEVEL**

TUGAS AKHIR

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

Oleh :

**Nafi' Kurniawan
NRP. 105 115 00000 064**

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr. Ir. Totok Soehartanto, DEA..... (Dosen Pembimbing I)
2. Sefi Novendra Patrialova, S.Si., MT..... (Dosen Pembimbing II)
3. Dr. Ir. Purwadi Agus Darwito, M.Sc..... (Dosen Penguji I)

SURABAYA
JULI 2018

**PERANCANGAN DAN PEMBUATAN *HUMAN MACHINE*
INTERFACE PADA SIMULATOR PENGENDALIAN
FLOW DAN LEVEL**

Nama Mahasiswa : Nafi' Kurniawan
NRP : 105 115 00000 063
Jurusan : D III Teknik Instrumentasi,
Departemen Teknik Instrumentasi FV-ITS
Dosen Pembimbing I : Dr. Ir. Totok Soehartanto, DEA
Dosen Pembimbing II : Sefi Novendra Patrialova, S.Si., MT

ABSTRAK

Dalam proses memonitoring sebuah kinerja pada sebuah plant atau simulator dibutuhkan integrasi antara perangkat keras dan perangkat lunak agar terwujudkan sebuah proses memonitoring yang telah diharapkan, untuk memberikan tampilan yang menggambarkan dinamika proses dalam sebuah plant atau simulator maka dibutuhkan *human machine interface* karena hal tersebut dapat menghasilkan sistem monitoring yang *user friendly* dengan itu dapat menjadi penghubung antara operator dengan plant atau simulator yang telah dikendalikan. Simulator ini dibuat untuk sarana pembelajaran tentang sistem pengendalian sehingga perlu adanya monitoring untuk memantau kinerja dari mesin tersebut. Metode penampilan data tidak hanya pada LCD melainkan juga pada *human machine interface* dengan variabel yang diamati ialah *flow* dan *level*. Dari pembacaan sensor akan direkam dengan menggunakan *localhost* dalam setiap detiknya menyimpan sebanyak 25 data.

Kata kunci : *Human machine interface, visual studio*

DESAIN OF HUMAN MACHINE INTERFACE ON FLOW AND LEVEL PLANT SIMULATOR

Name of Student : Nafi' Kurniawan
NRP : 105 115 00000 063
Departement : D III Teknik Instrumentasi,
Departemen Teknik Instrumentasi FV-ITS
Supervisor I : Dr. Ir. Totok Soehartanto, DEA
Supervisor II : Sefi Novendra Patrialova, S.Si., M.T.

ABSTRACT

In the process monitor a performance on a plant or simulator required integration between the hardware and the software to monitor a process that manifest had been expected, to give the appearance that illustrates the dynamics of the process in a plant or simulator required human machine interface because it can produce a user friendly monitoring systems with it can be a liaison between operators with plant or simulator that has been controlled. This Simulator is made for a means of learning about the control systems so that the need for monitoring to monitor the performance of the machine. A method of data not only appearance on the LCD but also on human machine interface with variable is the observed flow and level. From the readings of the sensors will be recorded using the localhost in every second of it keep as much as 25 data.

Keywords : Human machine interface, visual studio

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul **“PERANCANGAN DAN PEMBUATAN HUMAN MACHINE INTERFACE PADA SIMULATOR PENGENDALIAN FLOW DAN LEVEL”** dengan tepat waktu. terselesaikannya laporan ini juga tak luput dari dukungan dan peran serta dari orangtua dan keluarga besar serta berbagai pihak. Untuk itulah dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Dr .Ir. Purwadi Agus D, M.Sc. selaku Ketua Departemen Teknik Instrumentasi FV-ITS.
2. Bapak Dr. Ir. Totok Soehartanto, DEA dan Ibu Sefi Novendra Patrialova, S.Si., MT selaku pembimbing Tugas Akhir yang telah membina dengan baik dan sabar.
3. Bapak Detak Yan Pratama selaku Dosen Wali penulis.
4. Kedua orang tua yang tidak henti-hentinya memberi semangat dan doa untuk menyelesaikan Tugas akhir ini
5. Naimah selaku anggota *Flow and Level Demonstration Unit Team* yang telah berjuang bersama-sama dalam pengerjaan Tugas Akhir ini hingga selesai.
6. Teman-teman D3 Teknik Instrumentasi angkatan 2015 FV-ITS.
7. Serta semua pihak yang tidak dapat disebutkan satu persatu.

Demikian laporan Tugas Akhir ini penulis persembahkan dengan harapan dapat bermanfaat dalam akademik baik bagi penulis sendiri maupun bagi pembaca.

Surabaya, 20 Juni 2018

Penulis.

DAFTAR ISI

	Hal
HALAMAN JUDUL	i
LEMBAR PENGESAHAN.....	iii
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
BAB I. PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Permasalahan.....	2
1.3 Tujuan	2
1.4 Batasan Masalah.....	2
1.5 Manfaat	2
BAB II. TINJAUAN PUSTAKA	
2.1 Sistem Monitoring	3
2.2 Sistem Akuisisi Data	4
2.3 Komunikasi Serial	6
2.4 USART	8
2.6 <i>Human Machine Interface</i>	10
BAB III. PERANCANGAN DAN PEMBUATAN ALAT	
3.1 Bahan Dan Peralatan Yang Digunakan	11
3.2 Prosedure Penelitian	12
3.2.1 Diagram Alir	12

3.2.2 Diagram Blok Sistem HMI	14
3.2.3 Perancangan Perangkat Keras	15
3.2.4 Perancangan Perangkat Lunak.....	19
3.2.5 Pengujian Sistem HMI	23

BAB IV. HASIL DATA DAN PEMBAHASAN

4.1 Hasil Perancangan HMI	24
4.1.1 Perancangan Sistem HMI	25
4.1.2 Wirring Elektrik Pada Simulator	26
4.1.3 Hasil Pengujian Data Serial	27
4.1.4 Pengujian HMI	28
4.1.5 Pengujian Database	30
4.1.6 Analisa Pembacaan LCD & Database	31
4.1.7 Tampilan HMI.....	33
4.2 Pembahasan.....	33

BAB V. PENUTUP

5.1 Kesimpulan	45
5.2 Saran	45

DAFTAR PUSTAKA

BIODATA PENULIS

DAFTAR GAMBAR

	Hal
Gambar 2.1 Diagram Blok Sistem Pengukuran	3
Gambar 2.2 Diagram Blok Sistem Akuisisi Data	5
Gambar 2.3 Proses Akuisisi Data Berbasis Komputer	6
Gambar 2.4 Transmisi Serial	7
Gambar 2.5 Arah Transmisi Serial	7
Gambar 2.6 Register USART	10
Gambar 3.1 Diagram Alir Pengukuran	11
Gambar 3.2 Diagram Alir Pengerjaan Sistem	13
Gambar 3.3 Diagram Blok Sistem HMI.....	14
Gambar 3.4 Mikrokontroler Atmega 16	15
Gambar 3.5 Power Supply 12 V 10 A	16
Gambar 3.6 Buck Converter LM2596	17
Gambar 3.7 Komunikasi Serial FT232RL	17
Gambar 3.8 Kabel Mini USB to USB	18
Gambar 3.9 Komputer	19
Gambar 3.10 <i>Code Vision AVR</i>	20
Gambar 3.11 PuTTY	21
Gambar 3.12 VisualStudio 2017	22
Gambar 3.13 XAMPP	22
Gambar 3.14 HMI Pada Simulator Level	23

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di dalam sebuah kinerja proses yang prosesnya membutuhkan pengamatan yang akurat. Banyak sekali parameter-parameter yang perlu dimonitoring maupun dikontrol. Untuk memudahkan melakukan monitoring maupun kontrol tersebut, maka diperlukannya suatu sistem monitoring yang dapat diintegrasikan pada Simulator *flow* dan *level* karena simulator *flow* dan *level* yang beredar dipasaran masih belum dilengkapi sistem monitoring. (Hamburg, 2006) Menurut penelitian bahwa sistem monitoring adalah pemantauan yang dapat memberikan sebuah informasi dalam bentuk apapun untuk mendefinisikan status dari pemantauan yang dilakukan agar informasi dapat dikumpulkan dari hasil pemantauan maka dibutuhkan juga sebuah tempat untuk penyimpanan informasi agar dapat diamati dari informasi yang telah dikumpulkan (Putra & Febriani, 2013).

Dalam proses memonitoring sebuah kinerja pada sebuah plant atau simulator dibutuhkan integrasi antara perangkat keras dan perangkat lunak agar terwujudkan sebuah proses memonitoring yang telah diharapkan, untuk memberikan tampilan yang menggambarkan dinamika proses dalam sebuah plant atau simulator maka dibutuhkan *human machine interface* karena hal tersebut dapat menghasilkan sistem monitoring yang *user friendly* dengan itu dapat menjadi penghubung antara operator dengan plant atau simulator yang telah dikendalikan (Suryawati & Sustika, 2010).

Untuk itu pada tugas akhir ini dilakukan “Perancangan dan Pembuatan *Human Machine Interface* Pada Simulator Flow dan Level”. *Human machine interface* yang telah dirancang dan dibuat mampu untuk mendisplaykan dinamika sistem pengendalian proses pada system pengendalian level dan dapat digunakan untuk melakukan aksi melalui operator yang ada di depan *human machine interface*.

Dengan dibuatnya *human machine interface* maka memungkinkan pemakai simulator dapat mengamati dinamika proses dan kinerja pada sistem instrumen yang ada pada simulator tersebut.

1.2 Rumusan Permasalahan

Dari paparan latar belakang diatas maka permasalahan dari tugas akhir ini adalah bagaimana merancang dan membuat suatu unit simulator *flow* dan *level* yang bisa diintegrasikan dalam sebuah komunikasi *human machine interface*.

1.3 Tujuan

Tujuan dari penelitian ini adalah untuk merancang dan membuat sistem monitoring yang dapat ditampilkan pada *Human Machine Interface* dengan menggunakan VisualStudio sebagai perangkat lunak untuk mewujudkannya sehingga dapat dianalisa antara database dengan pembacaan pada LCD.

1.4 Batasan Masalah

Perlu diberikan beberapa batasan permasalahan agar pembahasan tidak menyimpang dari tujuan. Adapun batasan permasalahannya adalah sebagai berikut :

1. Perancangan dan pembuatan *human machine interface* dengan menggunakan *software* visual basic.
2. Variabel fisis yang ditampilkan pada HMI adalah *level*, *flow* dan prosentase bukaan *valve*.

1.5 Manfaat

Adapun manfaat yang dapat diperoleh dari tugas akhir ini adalah sebagai berikut :

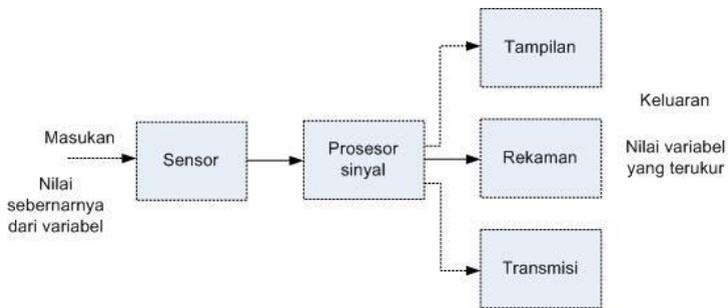
1. Sebagai media visualisasi tentang dinamika proses pada simulator *flow* dan *level*.
2. Sebagai sarana perekaman data pada kinerja sensor untuk mengamati pembacaan pada sensor.

BAB II TEORI PENUNJANG

2.1 Sistem Monitoring

Monitoring merupakan pemantauan pada suatu proses atau sistem yang mencapai hasil yang diharapkan baik secara kualitas dan kuantitas dengan efektif. Monitoring dilakukan dengan tujuan untuk meminimalkan gangguan jika terdeteksi suatu gangguan atau kegagalan dalam suatu sistem. Monitoring selain berfungsi sebagai pengawasan juga berfungsi untuk merekam apa yang terjadi pada sistem yang dimonitor dalam bentuk data table maupun grafik yang ditampilkan dalam bentuk *display* (Putra & Febriani, 2013).

Untuk mempermudah dalam melakukan pengukuran maka perlu diketahuimengenai pronsip dasar pengukuran seperti gambar berikut :



Gambar 2.1 Diagram Blok Sistem Pengukuran (Putra & Febriani, 2013)

Sensor adalah elemen sistem yang secara efektif berhubungan dengan proses dimana suatu variabel sedang diukur dan menghasilkan suatu keluaran dalam bentuk tertentu tergantung pada variabel masukannya, dan dapat digunakan oleh bagian sistem pengukuran yang lain untuk mengenali nilai variabel

tersebut. sebagai contoh adalah sensor termokopel yang memiliki masukan berupa temperatur serta keluaran berupa gaya gerak listrik (GGL) yang kecil. GGL yang kecil ini oleh bagian sistem pengukuran yang lain dapat diperkuat sehingga diperoleh pembacaan pada alat ukur.

Prosesor sinyal merupakan elemen sistem instrumentasi yang akan mengambil keluaran dari sensor dan mengubahnya menjadi suatu bentuk besaran yang cocok untuk tampilan dan transmisi selanjutnya dalam beberapa sistem kontrol. Seperti pengondisi sinyal (*signal conditioner*) merupakan salah satu bentuk prosesor sinyal.

Penampil data Elemen ini menampilkan nilai-nilai yang terukur dalam bentuk yang bisa dikenali oleh pengamat, seperti melalui sebuah alat penampil (*display*), misalnya sebuah jarum penunjuk (*pointer*) yang bergerak disepanjang skala suatu alat ukur. Selain ditampilkan, sinyal tersebut juga dapat direkam, misalnya pada kertas perekam diagram atau pada piringan magnetik, ataupun ditransmisikan ke beberapa sistem yang lain seperti sistem kontrol/kendali.

2.2 Sistem Akuisisi Data

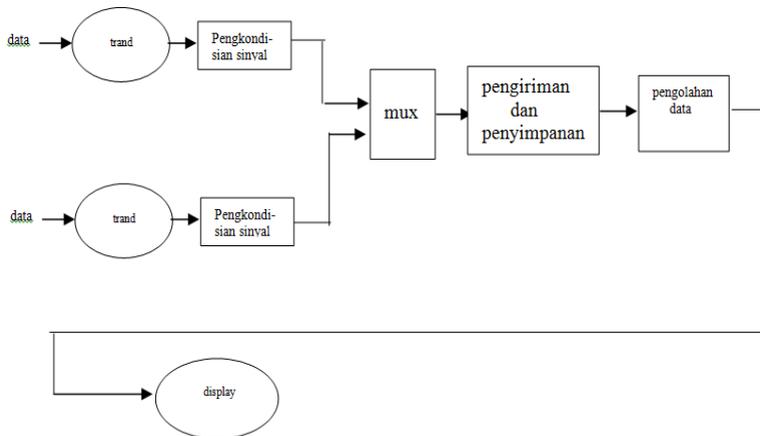
2.2.1 Definisi

Sistem adalah kumpulan komponen yang saling bekerja sama untuk mencapai tujuan tertentu. *Akuisisi data* merupakan sarana pengumpulan informasi. Biasanya, sistem akuisisi data dalam industri merupakan sistem yang *real time*. Sistem akuisisi data berfungsi sebagai antarmuka (*interface*) antara dunia nyata (*real world*) parameter fisik, yang analog, dengan dunia komputer, yang digital.

Sistem akuisisi data dapat didefinisikan sebagai suatu sistem yang berfungsi untuk mengambil, mengumpulkan dan menyiapkan data, hingga memprosesnya untuk menghasilkan data yang dikehendaki. Jenis serta metode yang dipilih pada umumnya bertujuan untuk menyederhanakan setiap langkah yang

dilaksanakan pada keseluruhan proses. Sistem akuisisi data merupakan sistem instrumentasi elektronik terdiri dari sejumlah elemen yang secara bersama-sama bertujuan melakukan pengukuran, menyimpan, dan mengolah hasil pengukuran. Semua besaran fisik yang akan diukur, diamati, disimpan, dan dikontrol dapat berupa suhu, tekanan, cahaya, suara, dll. Di *real world* besaran fisik itu dalam bentuk analog.

Suatu sistem akuisisi data pada umumnya dibentuk sedemikian rupa sehingga sistem tersebut berfungsi untuk mengambil, mengumpulkan dan menyimpan data dalam bentuk siap yang siap untuk diproses lebih lanjut. Gambar 2.2 menunjukkan diagram blok sistem akuisisi data.

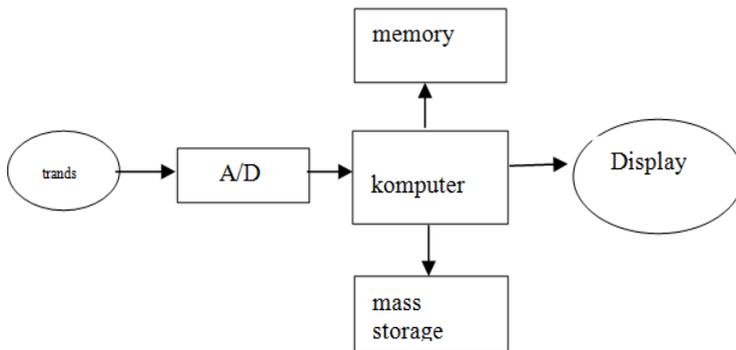


Gambar 2.2 Diagram Blok Sistem Akuisisi Data

2.2.2 Perkembangan Sistem Akuisisi Data

Pada mulanya proses pengolahan data lebih banyak dilakukan secara manual oleh manusia. Sehingga pada saat itu perubahan besaran fisis dibuat kebesaran yang langsung bisa diamati panca indra manusia. Selanjutnya dengan kemampuan teknologi pada bidang elektrikal besaran fisis yang diukur sebagai data dikonversikan ke bentuk sinyal listrik, data kemudian ditampilkan kedalam bentuk simpangan jarum, pendaran cahaya pada layar *monitor*, *recorder xy* dan lain-lain.

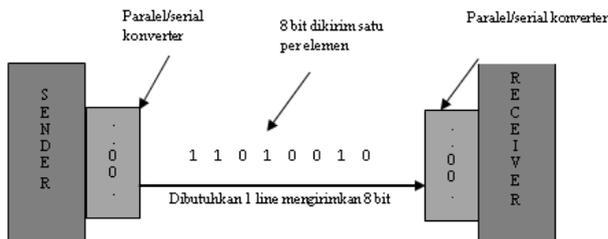
Sistem akuisisi data berkembang pesat sejalan dengan kemajuan dibidang teknologi digital dan komputer. Kini, akuisisi data menkonversikan besaran fisis data source ke bentuk sinyal digital dan diolah oleh suatu komputer. Pengolahan dan pengontrolan proses oleh komputer memungkinkan penerapan akuisisi data dengan *software*. *Software* memberikan harapan proses akuisisi data bisa divariasikan dengan mudah sesuai kebutuhan. Gambar 2.3 menunjukkan proses akuisisi data menggunakan komputer.



Gambar 2.3 Proses Akuisisi Data Berbasis Komputer

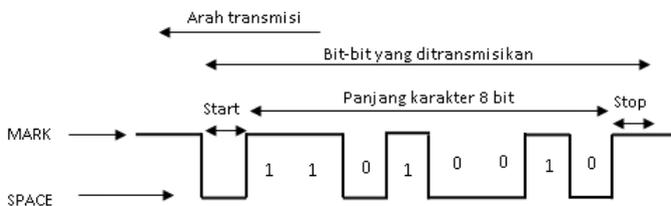
2.3 Komunikasi Serial

Komunikasi serial merupakan komunikasi yang pengiriman datanya dikirimkan per elemen. Dengan transmisi serial pengiriman data jarak jauh menjadi lebih efektif dibandingkan dengan transmisi paralel. Data paralel internal komputer dimasukkan ke pengubah paralel ke serial. Saluran serial mengirimkan setiap karakter per elemen sehingga hanya diperlukan satu atau dua penghantar, yaitu kirim data (TX) dan terima data (RX). Contoh transmisi serial ditunjukkan pada Gambar 1 (Suyadi, 2012).



Gambar 2.4 Transmisi Serial(Suyadi, 2012)

Pada Gambar 2.4 ditampilkan bahwa transmisi serial membutuhkan waktu yang relatif lebih lama dibandingkan dengan transmisi paralel, hal ini dikarenakan untuk komunikasi serial hanya menggunakan satu saluran. Berikut ini gambar arah transmisi dari komunikasi serial(Suyadi, 2012).



Gambar 2.5 Arah transmisi komunikasi serial(Suyadi, 2012)

Sebagai contoh, jika akan dikirim data serial 10011010, maka agar data tersebut dapat dikirim dan diterima dengan baik, selang waktu yang digunakan oleh pengirim dari penerima satu dengan yang lain harus sama. Jika penerima telah menerima penyesuaian bit, maka seharusnya juga harus segera menerima penyesuaian karakter, dan penerima juga harus mengetahui awal dan akhir blok data yang dikirim. Penyesuaian yang diperlukan dapat diperoleh secara sinkron maupun asinkron, data yang dikirim oleh terminal

komputer lewat jalur RX dimasukan ke pengubah seri ke paralel sebelum diteruskan ke komputer(Suyadi, 2012).

2.4 Komunikasi Data Serial USART (*Universal Synchronous Asynchronous Reciever Transmitter*)

Serial USART menggunakan metode *full duplex* (dua arah) antara *reeiver* dan *transmitter*. USART biasanya digunakan untuk komunikasi *asynchronous*. Artinya, tidak ada jam umum atau *clock* antara pemancar dan penerima untuk membuat mereka tetap sinkron satu sama lain. Untuk mempertahankan sinkronisasi antara pemancar dan penerima, framing start dan stop bits digunakan di awal dan akhir setiap byte data dalam urutan transmisi.

The USART Atmel juga memiliki fitur sinkron. Ketika menghubungkan komponen melalui *serial interface*, penting untuk memastikan parameter komunikasi antara pemancar dan penerima dicocokkan. Baik pemancar maupun penerima harus diatur untuk *Baud rate* yang sama, jumlah *start-bit* dan *stop-bit* yang sama, paritas, polaritas, level tegangan, dll

a. USART Transmitter

Pemancar USART terdiri dari *Transmit Shift Register*. Data yang akan dikirim dimuat ke dalam *Transmit Shift Register* melalui I/O USART *Data Register* (UDR). *Bit framing* mulai dan berhenti secara otomatis ditambahkan ke data dalam *Transmit Shift Register*. Paritas secara otomatis dihitung dan ditambahkan ke *Transmit Shift Register*. Data kemudian digeser keluar dari *Transmit Shift Register* melalui pin TxD sedikit demi sedikit pada *Baud rate* yang ditetapkan. Pemancar USART dilengkapi dengan dua bendera status: UDRE dan bendera TXC. Tanda USART Data Register Empty (UDRE) set ketika buffer transmisi kosong mengindikasikan siap menerima data baru. Bit ini harus ditulis ke nol ketika menulis USART *Control and Status Register A* (UCSRA). Bit UDRE dibersihkan dengan menulis ke I/O USART *Data Register* (UDR). *The Transmit Complete* (TXC) Flag bit diatur ke logika satu ketika seluruh frame di *Transmit Shift Register* telah digeser keluar, dan tidak ada data baru yang ada di *buffer*

transmisi. Bit TXC dapat disetel ulang dengan menulis logika untuknya.

b. USART Receiver

USART *receiver* secara virtual identik dengan USART *transmitter* kecuali arah aliran data dibalik. Data diterima sedikit demi sedikit melalui pin RxD di Baud Rate yang ditetapkan. *Receiver* USART dilengkapi dengan *Receive Complete* (RXC) Flag. Bendera RXC adalah logika satu ketika data yang belum dibaca ada di buffer penerima

c. USART Register

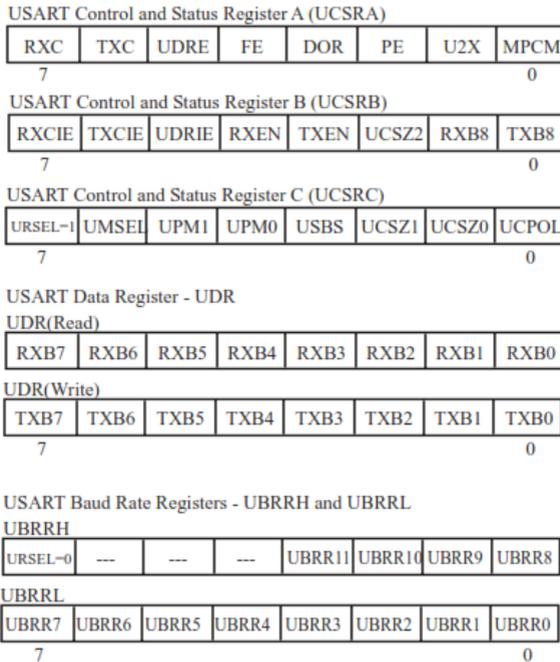
USART *Control and Status Register A* (UCSRA). Register UCSRA berisi bit RXC, TXC, dan UDRE. Fungsi bit-bit ini sudah dibahas.

USART *Control and Status Register B* (UCSRB). Register UCSRB berisi bit *Receiver Enable* (RXEN) dan bit *Transmitter Enable* (TXEN). Bit-bit ini adalah saklar "on / off" untuk penerima dan pemancar, masing-masing. Register UCSRB juga berisi bit UCSZ2. Bit UCSZ2 dalam register UCSRB dan bit UCSZ [1: 0] yang terdapat dalam daftar UCSRC bersama-sama mengatur ukuran karakter data.

USART *Control and Status Register C* (UCSRC). Daftar UCSRC memungkinkan pengguna untuk menyesuaikan fitur data ke aplikasi yang ada di tangan. Harus ditekankan bahwa baik pemancar maupun penerima dikonfigurasi dengan fitur data yang sama untuk transmisi data yang tepat. UCSRC berisi bit-bit berikut:

1. USART Mode Select (UMSEL) – 0 : operasi asynchronous, 1: operasi sinkron
2. USART Parity Mode (UPM[1:0]) - 00: tidak ada paritas, 10: bahkan paritas, 11: paritas ganjil
3. USART Stop Bit Select (USBS) - 0: 1 stop bit, 1: 2 stop bits

4. USART Character Size (data width) (UCSZ[2:0]) - 000: 5-bit, 001: 6-bit, 010: 7-bit, 011: 8bit, 111: 9-bit.



Gambar 2.6 Register USART

2.5 HMI (*Human Machine Interface*)

Human Machine Interface (HMI) adalah sistem yang menghubungkan antara manusia dan mesin. HMI dapat berupa pengendali dan visualisasi status, baik dengan manual maupun melalui visualisasi komputer yang bersifat real time. Dalam industri manufacture HMI dapat berupa suatu tampilan Graphic User Interface (GUI) pada layar monitor yang akan dihadapi oleh operator suatu mesin maupun pengguna yang membutuhkan data kerja mesin. HMI mempunyai kemampuan dalam hal visualisasi untuk monitoring dan data mesin yang terhubung secara online dan real time. HMI akan memberikan suatu gambaran kondisi mesin yang berupa peta mesin produksi di layar monitor dimana dapat dilihat bagian mesin mana yang sedang bekerja.

Selain itu pada HMI terdapat juga visualisasi pengendali mesin berupa push button, input reference dan sebagainya yang dapat difungsikan untuk mengendalikan mesin sebagaimana mestinya. Selain itu pada HMI dapat ditampilkan alarm jika terjadi kondisi bahaya di dalam mesin. HMI dapat juga menampilkan data-data rangkuman kerja mesin secara grafik.

Sistem HMI biasanya bekerja secara online dan real time dengan membaca data yang dikirimkan melalui I/O port yang digunakan oleh controller. Port yang biasanya digunakan untuk controller dan akan dibaca oleh HMI antara lain adalah port com, port USB, port RS232 dan ada pula yang menggunakan port serial.

Salah satu software HMI yang mudah digunakan untuk seorang pemula karena banyak tutorial di internet adalah visual studio, dengan kemudah yang diberikan oleh visual studio untuk memprogram, pengguna tidak perlu mengetik satu persatu kode – kode program untuk membuat sebuah interface meskipun tidak semua kode program ada pada *plugin* yang disediakan tetapi itu semua sangat membantu. (Suryawati & Sustika, 2010)

(Halaman Ini Sengaja Dikosongkan)

BAB III PERANCANGAN DAN PEMBUATAN ALAT

3.1 Bahan dan Peralatan Yang Digunakan



Gambar 3.1 Diagram Alir Pengukuran

Alat dan bahan yang digunakan dalam pengerjaan tugas akhir ini mengacu pada gambar 3.1 diagram blok sistem monitoring sehingga dibutuhkan alat dan bahan sebagai berikut:

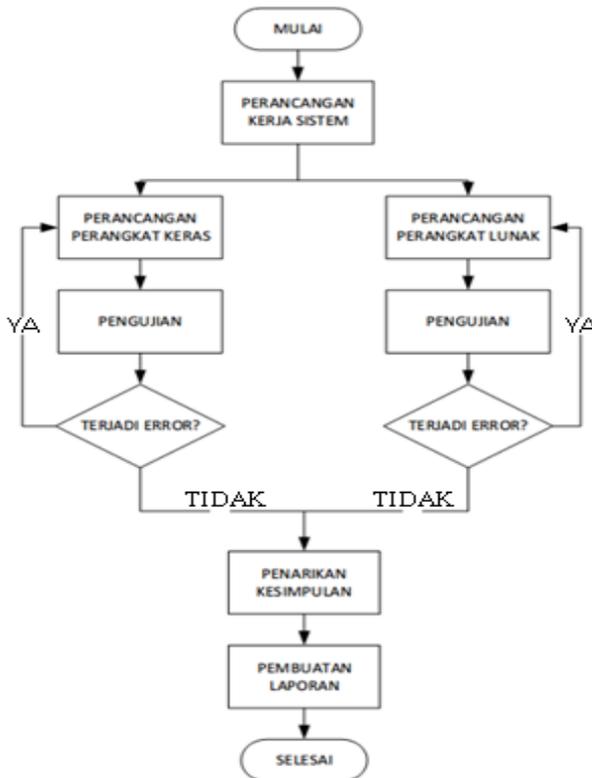
- Mikrokontroler Atmega 16
Dibutuhkan sebuah mikrokontroler sebagai media pemrosesan data dengan *input* berupa sensor yang dikondisikan sinyal keluarannya agar dapat diproses oleh mikrokontroler.
- *Power Supply* 12 volt
Dibutuhkan sumber daya untuk mencukupi kebutuhan listrik dalam seluruh komponen yang ada pada plant atau simulator.
- *Buck Converter* LM2596
Dibutuhkan sebuah penurun tegangan *DC to DC* agar sesuai dengan yang dibutuhkan oleh mikrokontroler karena tegangan yang dibutuhkan oleh mikrokontroler hanya sebesar 5 volt.
- Komunikasi Serial FT232RL
Dibutuhkan sebuah komponen untuk media pengiriman dan penerimaan data antara mikrokontroler dengan komputer.
- Kabel *Mini USB to USB*
Dibutuhkan kabel penghubung antara FT232RL dengan komputer karena keluaran dari FT232RL berupa *Mini USB*.

3.2 Prosedur Penelitian

Agar didapatkan hasil yang sesuai maka dibutuhkan metode dalam pengerjaan tugas akhir ini digambarkan pada diagram alir perancangan alat mulai dari perangkat keras dan perangkat lunak.

3.2.1 Diagram Alir Perancangan Alat

Diagram dibawah ini menjelaskan tentang bagaimana merancang *Human Machine Interface* pada simulator flow dan level.



Gambar 3.2 Diagram Alir Pengerjaan Sistem

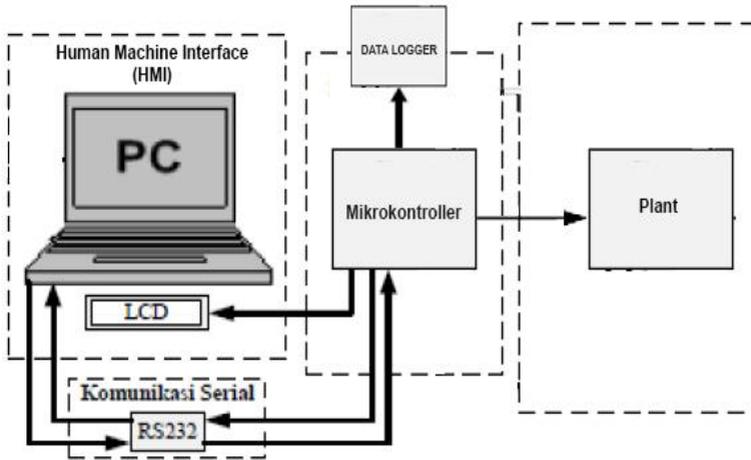
Pada diagram alir tersebut menggambarkan tentang perancangan sebuah sistem pada simulator flow dan level yang

mempunyai perangkat keras dan perangkat lunak dimana saling terintegrasi satu sama lain agar tercipta sistem yang dapat dikendalikan. Adapun langkah yang harus di kerjakan adalah sebagai berikut :

1. Perhitungan Dimensi dan Perancangan sistem *Hardware dan Software*. Pada tahap ini dilakukan perhitungan dimensi serta spesifikasi dari Tangki pengukuran level yang dibutuhkan untuk acuan spesifikasi pada pompa agar dapat mengisi hingga setpoint dengan waktu yang tidak terlalu lama. Pada tahap ini juga dilakukan perancangan sistem *hardware* agar pembacaan sensor dapat dikirim melalui serial port dan ditampilkan pada *HMI* dan perancangan perangkat lunak untuk menampilkan data yang berada pada serial port.
2. Pembuatan Alat
Tahapan ini yaitu dilakukan perakitan atau penggabungan komponen-komponen yang dibutuhkan berupa *power supply, controller, komunikasi serial, dan display(interface)*
3. Integrasi *Hardware, Software, dan Pengujian Alat*
4. Pada tahapan ini alat yang sudah dirakit serta program yang telah dibuat diuji dengan dialiri debit yang sesuai untuk mengetahui apakah alat yang sudah dibuat dapat berfungsi sebagaimana yang diinginkan.

3.2.2 Diagram Blok Sistem *Human Machine Interface*

Diagram dibawah ini ialah diagram sistem kerja *Human Macine Interface* mulai data pembacaan sensor yang diproses oleh mikrokontroller dan data dari sensor dapat menjadi acuan untuk aktuator dalam melakukan pengkondisian agar sistem berjalan dengan stabil. Untuk memantau apakah plant ini stabil dibutuhkan sebuah perekaman data agar dapat mengetahui dinamika proses pada plan tersebut



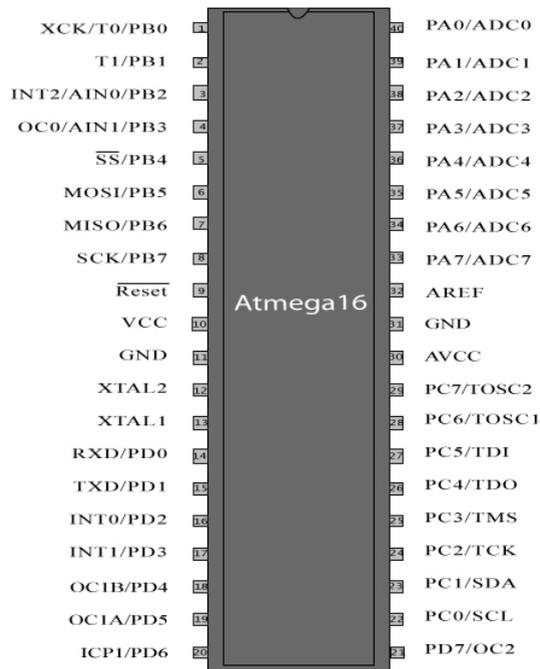
Gambar 3.3 Diagram Blok Sistem *Human Machine Interface*

Sebagai acuan dalam pembuatan *human machine interface* agar data pada mikrokontroller dapat dikirim ke komputer dibutuhkan penghubung antara komputer dengan mikrokontroller yaitu sebuah komunikasi serial berupa komponen FT232RL dan data dari mikrokontroller dikirim melalui komunikasi data serial USART dengan mengandalkan pin RX (penerima) dan TX (pengirim).

3.2.3 Perancangan Perangkat Keras

- Mikrokontroller Atmega 16

Gambar dibawah ini menggambarkan tentang pin – pin yang tersedia oleh mikrokontroller atmega 16 agar sesuai dengan kebutuhan maka harus dianalisa terlebih dahulu tentang port-port yang tersedia dan memperhitungkan dari berapa banyak sensor yang digunakan dan aktuator yang digunakan agar sesuai kebutuhan lebih mengarah pada keefisiensinya. Maka sangatlah penting untuk mendesign semua kebutuhan plant terlebih dahulu, untuk mengatasi kesalahan pembelian komponen dan kesalahan dalam pemasangan.



Gambar 3.4 Mikrokontroler Atmega 16

Atmega 16 merupakan salah satu produk dari perusahaan Atmel, yaitu mikrokontroler AVR 8bit. ATmega16 sebenarnya mempunyai spesifikasi yang sama dengan ATmega32 dan ATmega8535, perbedaannya hanya terletak pada besarnya kapasitas flash memori. ATmega16 memiliki kapasitas flash memori 16KByte, sedangkan ATmega32 mempunyai kapasitas memori 32KByte dan ATmega8535 mempunyai 8KByte.

- ***Power Supply 12 Volt 10 Ampere***

Gambar dibawah ini sebagai komponen untuk memenuhi kebutuhan sumber daya listrik.

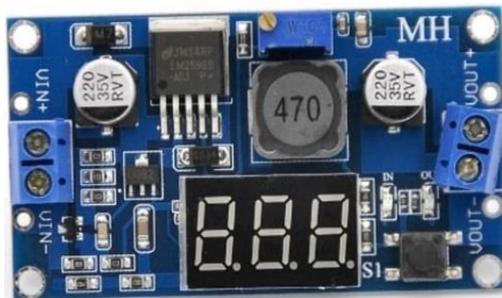


Gambar 3.5 *Power Supply 12 Volt 10 Ampere*

Power Supply digunakan untuk memenuhi kebutuhan listrik pada plant dengan merubah arus AC (*alternating current*) menjadi arus DC (*direct current*) karena pada plant membutuhkan arus DC (*direct current*) sebagai sumbernya.

- ***Buck Converter LM2596***

Gambar dibawah ini adalah komponen penurun tegangan agar sesuai dengan kebutuhan.

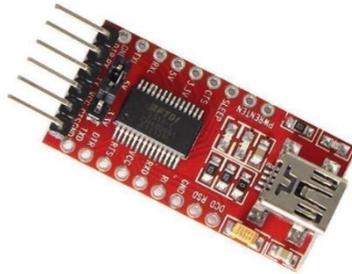


Gambar 3.6 *Buck Converter LM2596*

LM2596 adalah komponen penurun tegangan karena sumber listrik yang ada menggunakan tegangan 12 *volt* maka harus disesuaikan dengan kebutuhan pada mikrokontroller yang membutuhkan tegangan 5 *volt*.

- **Komunikasi serial FT232RL**

Gambar dibawah ini adalah media komunikasi serial dengan output berupa *Mini USB*



Gambar 3.7 Komunikasi serial FT232RL

Sebagai komunikasi data antara mikrokontroller dengan komputer dengan mengandalkan pin RX, TX, VCC, GND yang menghubungkan FT232RL pada port D0 untuk RX dan Port D1 untuk TX di port yang tersedia oleh mikrokontroller atmega 16.

- **Kabel *Mini USB to USB***

Gamba dibawah ini adalah sebuah kabel yang menghubungkan FT232RL dengan komputer.



Gambar 3.8 Kabel *Mini* USB to USB

Dibutuhkan kabel *mini* USB to USB karena *output* dari FT232RL ialah sebuah *mini* USB dan konektor yang tersedia pada komputer ialah USB.

- Komputer

Gambar dibawah ini adalah sebuah komputer untuk media pemrograman.



Gambar 3.9 Komputer

Dibutuhkan sebuah komputer sebagai media untuk membuat sebuah program yang nantinya akan digunakan untuk program dari mikrokontroller.

3.2.4 Perancangan Perangkat Lunak

- CodeVision AVR

Sebagai pembuat program USART agar terjadi sebuah komunikasi *full duplex* (dua arah) antara pengirim data dengan penerima data.

```
printf ("%d;", level);
lcd_gotoxy(0,0);
lcd_putsf("Level = ");
sprintf(temp,"%2d cm      ",level);
lcd_puts(temp);

if((periode==0)&&(minperiode==0))frek=0; //flow
frek=1/((periode*0.000128)+(minperiode*0.0000005));

flow=frek*0.13;//dikali dengan hasil kalibrasi

lcd_gotoxy(0,1);
lcd_putsf("Flow =");//tampilan lcd
sprintf(temp,"%4d flow",flow);
lcd_puts(temp);
printf("%d", flow);

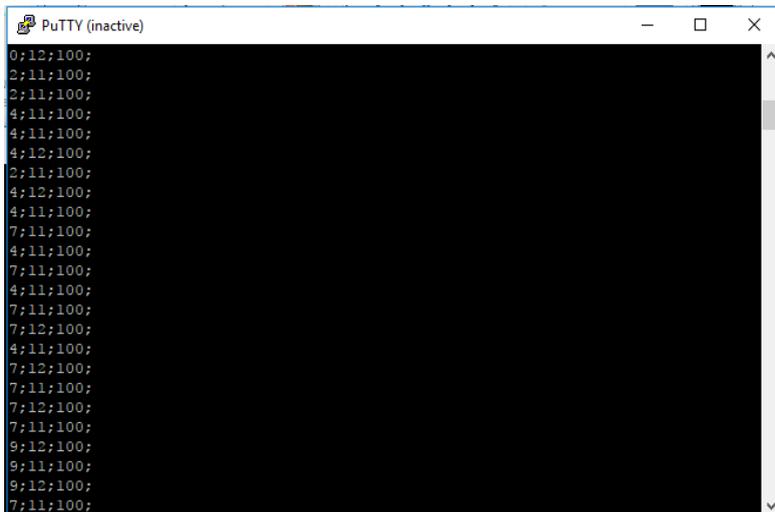
valve_open=stepcount/(maxstep/100);
lcd_gotoxy(0,2);
lcd_putsf("Valve = ");
sprintf(temp,"%3d %c",valve_open,'%');
printf("%d;\r\n",valve_open);
lcd_puts(temp);
```

Gambar 3.10 CodeVision AVR

Adanya perintah `printf` maka dapat mengirim data pada USART dengan pemisah data berupa “;”

- PuTTY

Aplikasi yang satu ini berfungsi untuk menampilkan jendela terminal seperti perintah “`cmd`” pada windows agar mengetahui apakah format yang dikirim USART melalui serial port telah sesuai dengan yang diinginkan

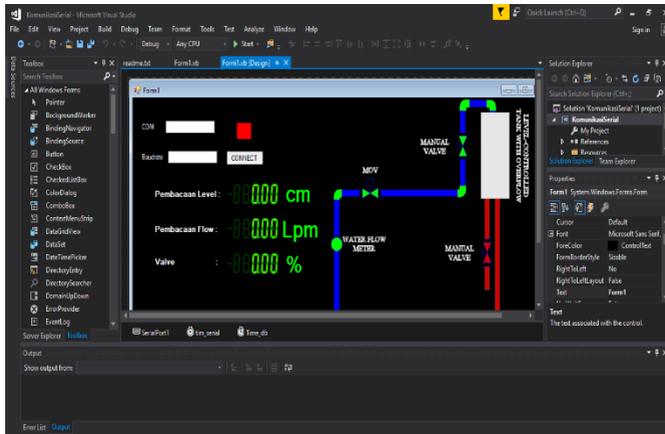


```
PuTTY (inactive)
0;12;100;
2;11;100;
2;11;100;
4;11;100;
4;11;100;
4;12;100;
2;11;100;
4;12;100;
4;11;100;
7;11;100;
4;11;100;
7;11;100;
4;11;100;
7;11;100;
7;12;100;
4;11;100;
7;12;100;
7;11;100;
7;12;100;
7;11;100;
9;12;100;
9;11;100;
9;12;100;
7;11;100;
```

Gambar 3.11 Terminal PuTTY

- VisualStudio 2017

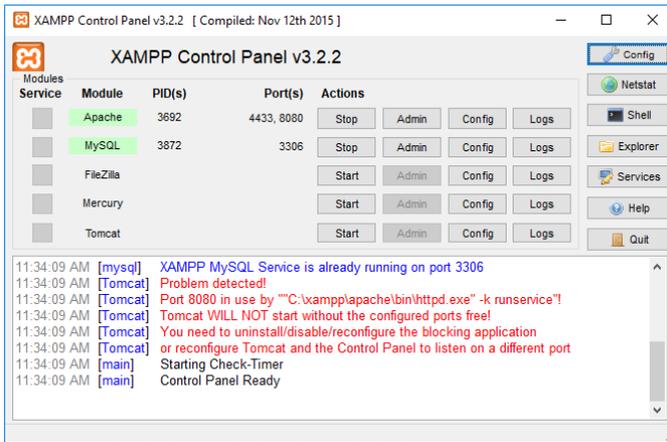
VisualStudio ialah sebuah perangkat lunak yang telah digunakan untuk membuat program untuk *human machine interface* yang mana menggunakan bahasa VB.net dalam membuat programnya



Gambar 3.12 VisualStudio 2017

- XAMPP

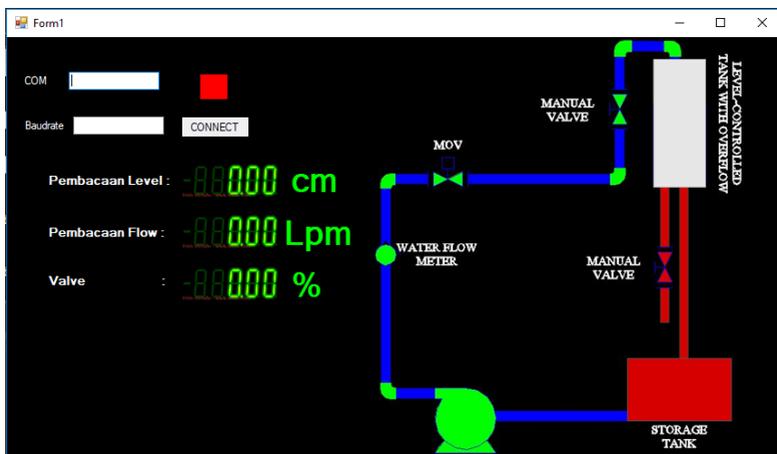
Aplikasi ini berfungsi untuk membuat database atau penyimpanan data yang telah ditampilkan oleh *human machine interface*.



Gambar 3.13 XAMPP Contro Panel v3.2.2

3.2.5 Pengujian Sistem *Human Machine Interface*

Tampilan *human machine interface* ini dibuat dengan bantuan software VisualStudio 2017 yang dapat terhubung dengan localhost untuk database atau penyimpanan data yang telah ditampilkan pada bagian pembacaan level, pembacaan flow, valve. Saat memulai program diwajibkan mengetahui “com” pada serialport yang terhubung dengan membukan device manager dan pada bagian ports dibuka. Setelah mengetahui “com” maka dilanjutkan memasukkan baudrate yang telah ditentukan pada saat membuat *human machine interface*



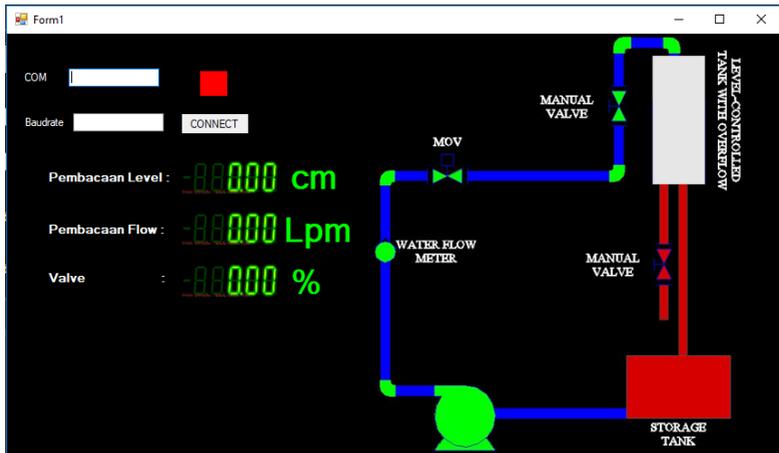
Gambar 3.14 *Human Machine Interface* Pada Simulator Level

(Halaman Ini Sengaja Dikosongkan)

BAB IV ANALISA DATA DAN PEMBAHASAN

4.1 Hasil Perancangan dan Pembuatan *Human Machine Interface*

Berikut ini adalah hasil dari Perancangan dan Pembuatan *Human Machine Interface*



Gambar 4.1 Perancangan dan Pembuatan *Human Machine Interface*

Pada gambar diatas adalah hasil dari perancangan dan pembuatan *human machine interface* untuk media visualisasi pada simulator pengendalian level. Didalam HMI yang telah dibuat terdapat beberapa komponen seperti serialport, textbox, label, button, picturebox dan plugin dari MitovLab. Untuk fungsi dari penggunaan komponen serialport adalah sebagai jalur untuk komuikasi data mulai dari mikrokontroller sampai dapat terbaca oleh komputer. Fungsi komponen textbox adalah sebagai media memberi masukan sebuah data, jadi textbox sebenarnya tidak jauh beda dengan label hanya yang membedakan ialah jika textbox

nilainya dapat dirubah – rubah sedangkan label tidak bisa hanya seperti stiker yang telah di design dan di tempelkan. Untuk penggunaan textbox pada label COM itu berfungsi untuk memasukkan data COM yang terbaca pada komputer dengan cara membuka device manager dan pada port akan terlihat pada COM berapa, sedangkan untuk textbox pada bagian label BaudRate ialah nilai yang dimasukkan berupa nilai 9600 darimana asal nilai tersebut ialah berasal dari standart kecepatan pengiriman data, dapat kita rubah semau kita sesuai yang telah kita atur pada pengaturan serial port. Pada tampilan HMI terlihat gambar tentang plant yang sedang diamati, untuk mewujudkannya dengan menggunakan perintah picturebox untuk sebelumnya kita mendesign terlebih dahulu tentang bagaimana tampilan yang akan disajikan pada *interface* HMI. Oleh karena itu dibutuhkan HMI untuk penghubung antara operator dengan plant secara *realtime*, sehingga dapat memudahkan tugas sebagai operator untuk mengamati kinerja dari plant tersebut.

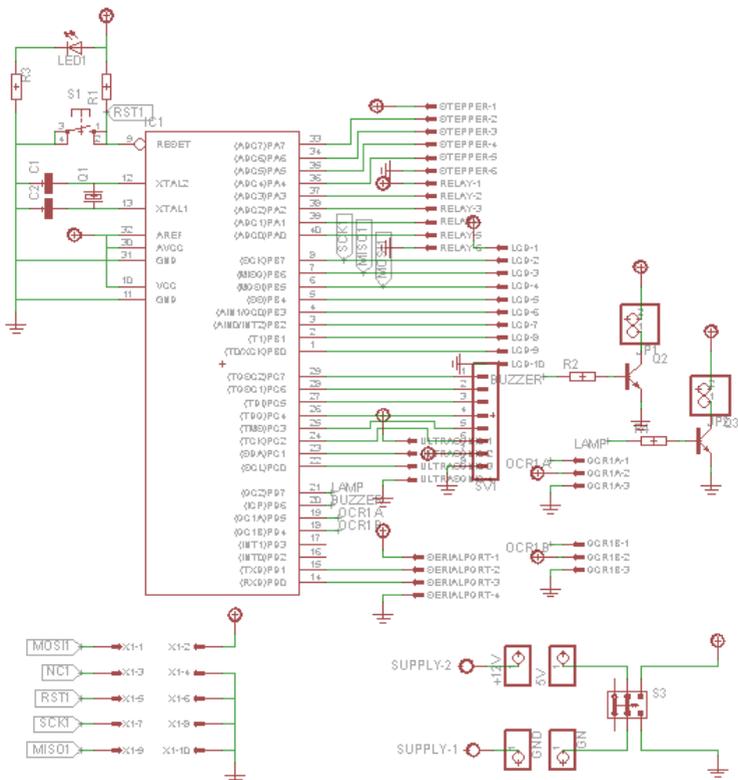
4.1.1 Perancangan Sistem *Human Machine Interface*

Perancangan sistem HMI ini dilakukan dengan merancang sebuah sistem monitoring untuk media mengamati kinerja dan dinamika proses pada plant agar dapat diamati kinerja pada plant tersebut maka perlu adanya penyimpanan sebuah data karena dengan melakukan penyimpanan data maka dapat diamtai bagaimana proses pada saat plant ini bekerja apakah ada *overshoot* atau *lowershoot* ataupun error sehingga mengakibatkan *safety* pada plant tersebut menyala semua dapat diketahui dengan mengamati dari data yang tersimpan, untuk media penyimpanan maka dibutuhkan pula tempat untuk menyimpan data tersebut, data yang telah ditampilkan pada tampilan HMI dapat tersimpan pada localhost dengan variabel yang direkam adalah *level*, *flow* dan bukaan valve. Data pembacaan level didapatkan dari pembacaan sensor ultrasonik dengan menggunakan suara ultrasonik sebagai cara kerjanya yaitu dengan pantulan dari suara ultrasonik pada media tertentu maka didapatkan waktu yang berbeda-beda

tergantung jarak dari media pemantulnya, untuk pembacaan debit air didapatkan dari sensor Flowmeter dengan cara kerja elektromagnetik yang didapat dari putaran baling-baling yang memutar sehingga didapatkan sebuah pulsa-pulsa tertentu tergantung kecepatan putarnya dan pembacaan bukaan valve didapatkan dari step yang dikeluarkan oleh mikrokontroler untuk memerintahkan stepper berputar membuka atau menutup valve. Penjelasan ini tentang penjelasan perancangan sebuah sistem HMI untuk mengetahui kinerja pada plant. Pada perancangan perangkat keras dilakukan perancangan mulai dari mikrokontroler yang digunakan dan mikrokontroler yang digunakan beracuan pada komponen yang digunakan untuk membuat plant tersebut, mulai dari sensor sebagai masukan data untuk mikrokontroler dan untuk keluaran yang digunakan berupa komponen apa saja. Sedangkan pada perancangan perangkat lunak yang digunakan juga beracuan pada mikrokontroler yang digunakan dan perangkat lunak yang mudah dimengerti, tentang fitur-fitur yang mudah digunakan yaitu salah satu contohnya ialah VisualStudio karena kemudahan fitur yang dapat digunakan maka banyak pemula yang menggunakan *software* tersebut sehingga artikel-artikel dalam dunia internet mudah dijumpai sehingga tidak membutuhkan waktu yang lama untuk melakukan pembuatan HMI. Dari hasil perancangan sistem didapatkan spesifikasi tentang *Hardware* dan *Software* yang dibutuhkan dalam pembuatan HMI.

4.1.2 Wiring Elektrik Pada Simulator

Wiring adalah alur antar komponen yang saling terhubung dengan menggunakan kabel yang menghubungkan pada Mikrokontroler. Dibutuhkan perencanaan sistem untuk membuat sebuah plant sebelum kita membangunnya dengan menggunakan *software* PROTEUS untuk mensimulasikan hasil dari perancangan sistem pada sebuah plant, karena tujuan dari merencanakan segala sesuatunya maka dapat meminimalisir waktu dan kesalahan dalam pembelian komponen sehingga mengetahui jalur elektrik pada setiap komponen



4.1.3 Hasil Pengujian Data Serial

Hasil dari pengujian data serial dengan mengamati proses pengiriman data sebelum ditampilkan pada HMI maka harus dilakukan pengujian tentang format yang terlihat pada aplikasi PuTTY.

```

PuTTY (inactive)
0:12:100;
2:11:100;
2:11:100;
4:11:100;
4:11:100;
4:12:100;
2:11:100;
4:12:100;
4:11:100;
7:11:100;
4:11:100;
7:11:100;
4:11:100;
7:11:100;
7:12:100;
4:11:100;
7:12:100;
7:11:100;
7:12:100;
7:11:100;
9:12:100;
9:11:100;
9:12:100;
7:11:100;

```

Gambar 4.3 Data Pembacaan Serial

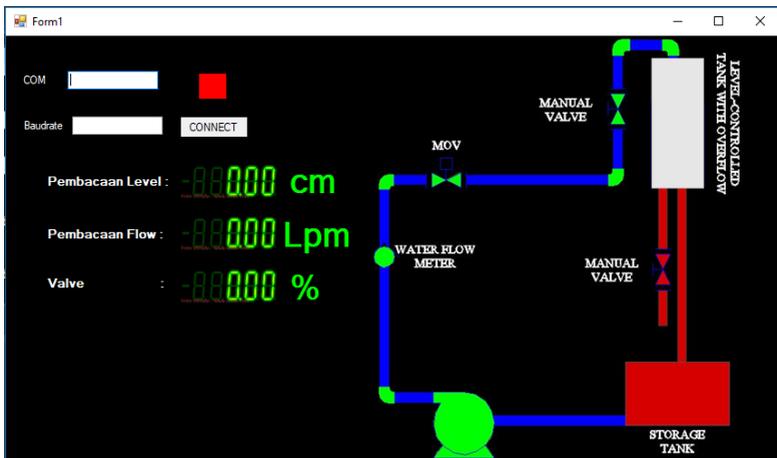
Data pembacaan sensor selanjutnya dikirim menggunakan komunikasi serial USART, pada ATmega 16 data pembacaan sensor dikirim menggunakan USART0, pengujian data serial yang terkirim dari Atmega16 menuju PC dipastikan pembacaannya menggunakan PuTTY.

Pada pengiriman data terdapat sekat berupa tanda “;” yang berfungsi sebagai pembeda antara data pembacaan flow, level, dan valve agar pembacaan sensor dapat dipisah menggunakan fitur split yang terdapat pada Vb.Net.

4.1.4 Pengujian HMI (*Human Machine Interface*)

Untuk menguji kinerja dari HMI maka dilakukan percobaan dengan menjalankan plant yang telah dibuat dengan melakukan percobaan pengisian air dari tempat penyimpanan air,

air dipompa dengan menggunakan 2 pompa dan pada *control valve* dalam keadaan membuka penuh karena *control valve* menerima perintah dari mikrokontroller karena *level* yang terbaca masih pada kondisi kosong, air terus dipompa sampai setpoint (20 cm) pada saat mikrokontroller membaca bahwa *level* melebihi setpoint maka memerintahkan *control valve* untuk mengurangi debit air yang masuk pada tangki pengukuran dan pada saat *level* kurang dari setpoint maka *control valve* akan membuka dan begitu seterusnya karena *control valve* selalu menyesuaikan setpoint. Untuk mengetahui debit dan ketinggian air maka terdapat sebuah LCD untuk media monitoring pada plant, sementara untuk HMI perlu menyambungkannya terlebih dahulu dengan cara menghubungkan kabel USB ke port USB komputer dan jangan lupa mengisi textbox COM dan BaudRate untuk menunjukkan bahwa plant dan HMI telah terhubung maka label yang berwarna merah akan berubah warna menjadi hijau. Setelah itu dapat dibandingkan data pembacaan pada LCD dengan HMI dan disajikan pada tabel 4.1.6 untuk desain dari HMI pada plan adalah sebagai berikut.



Gambar 4.4 Human Machine interface

Pada gambar tersebut terdapat tiga label yang menunjukkan pembacaan dari masing-masing sensor dan data yang diterima selanjutnya dipisah menggunakan fitur split pada Vb.Net.

```

0 references
27 Private Sub tim_serial_Tick(sender As Object, e As EventArgs) Handles tim_serial.Tick
28     SegmentGauge1.Text = data_masuk
29     Try
30         Dim data As String() = data_masuk.Split(";") '123;444;135;122;
31         SegmentGauge1.Value = data(0)
32         SegmentGauge2.Value = data(1)
33         SegmentGauge3.Value = data(2)
34         ProgressBarVertical1.Value = data(0)
35     Catch ex As Exception
36
37     End Try
38 End Sub

```

Gambar 4.5 Fungsi Split pada Vb.Net

Dari gambar diatas adalah fungsi yang digunakan oleh Vb.Net untuk memanggil setiap data yang telah dilakukan split dengan merubah urutan mulai dari 0 sampai seterusnya secara berurutan.

4.1.5 Pengujian Database

Pengujian database database pada localhost akan diuji dengan pembacaan pada LCD.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	level	varchar(6)	latin1_swedish_ci		No	None			Change Drop
2	flow	varchar(6)	latin1_swedish_ci		No	None			Change Drop
3	valve	varchar(6)	latin1_swedish_ci		No	None			Change Drop
4	time	timestamp(6)		on update CURRENT_TIMESTAMP	No	0000-00-00 00:00:00.000000		ON UPDATE: CURRENT_TIMESTAMP	Change Drop

Gambar 4.6 Database menggunakan MySQL

Pembacaan data pada HMI juga tersimpan pada sebuah *database* menggunakan MySQL dengan membuat sebuah *server*

localhost dapat digunakan sebagai tempat penyimpanan data pembacaan sensor dengan disertakan waktu *record*. Pertama adalah pembuatan *server* localhost yang dibuat melalui browser. Pada database yang dibuat terdapat 4 kolom data yang masing-masing akan menjadi tempat penyimpanan dari setiap pembacaan sensor yang telah dipisahkan menggunakan fitur yang telah dijelaskan sebelumnya. Dan untuk kolom 4 adalah tambahan untuk waktu record data, setelah database telah dibuat. Langkah selanjutnya adalah menginisiasi alamat dari database pada Vb.Net

```

46 Private Sub Time_db_Tick(sender As Object, e As EventArgs) Handles Time_db.Tick
47 Dim cmd As MySqlCommand
48 con.Open()
49 Try
50     cmd = con.CreateCommand
51     cmd.CommandText = "INSERT INTO `database`
52     VALUES
53     (@level, @flow, @valve, "" & Format(Date.Now, "yyyy-MM-dd HH:mm:ss") & "")"
54     cmd.Parameters.AddWithValue("@level", SegmentGauge1.Value)
55     cmd.Parameters.AddWithValue("@flow", SegmentGauge2.Value)
56     cmd.Parameters.AddWithValue("@valve", SegmentGauge3.Value)
57
58     cmd.ExecuteNonQuery()
59     con.Close()
60
61 Catch ex As Exception
62     MsgBox(ex.ToString)
63 End Try
64 con.Close()
65 End Sub
66

```

Gambar 4.7 Fungsi write pada database

Setelah mendeklarasikan alamat database pada Vb.Net, kemudian ditambahkan fungsi untuk menuliskan data pembacaan pada HMI agar tersimpan pada database.

4.1.6 Analisa Pembacaan LCD dan Database

Dari data yang telah diproses dan telah tersimpan pada database dapat dianalisa dengan cara membandingkan antara pembacaan sensor yang ditampilkan pada LCD dengan yang tersimpan pada database karena adanya perbedaan sitem pengiriman data antara LCD menggunakan pengiriman data secara paralel dan pada HMI pengiriman data secara serial, dari konsep tersebut dijelaskan bahwa pengiriman data secara paralel lebih cepat dibandingkan dengan pengiriman data secara serial.

Tabel 4.1 Pengujian Level Pada LCD Dengan Database

Data ke-	LCD	HMI	Data ke-	LCD	HMI
1	2	0	11	6,5	6,5
2	2	2	12	7,1	7,1
3	2	2	13	7,7	7,7
4	2,6	2,6	14	8,2	8,2
5	2,9	2,9	15	8,8	8,8
6	3,5	3,5	16	9,4	9,4
7	4,1	4,1	17	10	10
8	4,7	4,7	18	10,6	10,6
9	5,2	5,2	19	11,2	11,2
10	5,8	5,8	20	11,8	11,8

Dari tabel dijelaskan bahwa pembacaan pada LCD dengan yang tersimpan pada HMI adalah sama. Karena variabel yang diproses sedikit dan data yang dikirim melalui serial juga sedikit. Namun semua itu dipengaruhi oleh konsep dasar dari pengiriman data serial yang mana setiap waktu atau detak *clock* hanya dikirimkan 1 bit saja, bit dikirim satu persatu dengan satu buah jalur. Oleh karena itu kecepatan data pada transmisi data serial biasanya akan lebih lambat jika dibandingkan dengan transmisi data paralel. Transmisi data paralel adalah sebuah konsep yang digunakan oleh LCD untuk menampilkan sebuah karakter.

4.2 Pembahasan

Dilakukannya pembuatan *human machine interface* pada simulator flow dan level agar dapat ditampilkan pada layar komputer dan komunikasi yang digunakan adalah full duplex karena data dikirimkan dalam dua arah secara bersamaan antara pengiriman data atau penerimaan data. Dengan menggunakan

FT232RL dapat mewujudkan komunikasi tersebut. Pada komponen tersebut terdapat empat pin yaitu terdiri dari pin VCC, GND, RX, TX. Untuk pin tersebut terhubung pada port D0 dan port D1. Untuk data yang diperoleh hanyalah database saja karena HMI ini hanya memberikan sebuah perintah dalam bahasa c pada mikrokontroler agar dapat mengirimkan data dan diterima oleh visual studio dengan melalui serial port, data yang dikirim melalui serial port adalah data yang telah ditambahkan perintah “printf” pada function program untuk setiap pembacaan sensor, yang bertujuan untuk “split” data dengan pemisah data berupa “;” agar terkirim pada USART dengan format yang telah diinginkan.

Setelah itu digunakan program PuTTY untuk menguji apakah telah sesuai dengan format yang telah disiapkan dengan menggunakan pemisah “;” untuk membatasi data level, flow, dan valve, setelah itu data yang dikirim oleh USART dibaca oleh visual studio dengan menggunakan “split” data untuk pembagian data mana yang akan dibaca, dengan fungsi data(0) adalah data bagian awal baris yang ditampilkan oleh terminal pada PuTTY, data(1) untuk disamping data(0) dengan pemisah “;” contohnya 000;111;222;333 jadi ada sebuah pemisah setiap data adalah “;”. Untuk menambahkan variabel yang akan ditampilkan pada HMI maka perlu ditambahkan variabel pada USART dengan menambahkan perintah “printf” agar terkirim pada serial port.

Transmisi data yang digunakan oleh komunikasi serial adalah komunikasi yang setiap detak *clock* dikirimkan 1 bit saja dengan menggunakan satu jalur. Oleh karena itu transmisi data serial relatif lebih lambat dibandingkan dengan transmisi data paralel atau transmisi data yang digunakan oleh LCD untuk menampilkan setiap karakter yang telah direncanakan.

Oleh karena itu tidak banyak data yang dapat dibahas karena HMI ini bertujuan untuk mengidentifikasi saat terjadinya error pada sebuah sistem pengendalian agar mengetahui apa penyebabnya dan bagaimana tindakannya karena *human machine interface* dalam dunia industri diletakkan pada *control room* dan ada salah satu operator yang mengawasinya.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan rancang bangun dan analisa data yang telah dilakukan maka dapat disimpulkan bahwa :

1. Program Visual Studio dapat menjadi tampilan sebagai *monitoring plant* dan dapat menunjukkan perubahan setpoint ataupun memberikan inputan pada akuator
2. Komunikasi Serial dapat digunakan sebagai penghubung antara mikrokontroller dengan komputer
3. Hasil yang didapat oleh data logger menunjukkan bahwa sensor ultrasonic tidak memberikan penunjukan level secara stabil karena dipengaruhi oleh gelombang air yang tidak stabil

5.2 Saran

Adapun saran untuk penelitian selanjutnya agar hasil yang dicapai dapat memenuhi harapan adalah sebagai berikut :

1. Memastikan bahwa plant sudah tidak ada yang bocor untuk menghindari *losse* saat pengendalian karena akan mempengaruhi laju aliran dan membuat pengukuran tidak akurat dan presisi.
2. Sensor ultrasonik tidak cocok untuk pengukuran level dengan air yang terombang – ambing dan air ya turung dari atas di sebelah sensor ultrasonik. Karena sensor ultrasonik berdasarkan cara kerjanya adalah mengirimkan suara ultrasonik dan dipantulkan pada benda yang terkena pancaran suaranya sehingga diterima kembali oleh sensor ultrasonik

(Halaman Ini Sengaja Dikosongkan)

DAFTAR PUSTAKA

- [1] Bakti, Dheka, dkk. (2013). *PENGENDALIAN LEVE MENGGUNAKAN PENGENDALI PROPORSIONAL BERBASIS ATMEGA*. Surabaya: ITS.
- [2] WingWijayanto. (2012). *RANCANG BANGUN SISTEM PENGENDALIAN LEVEL DAN FLOW*.
- [3] Punchstein, A.F., Lioyd,T.C., Conrad, A.G., 1960. "Alternating Current Machines". New York: John Willey and Sons Inc.
- [4] Anonim. 2016. *Modul Praktikum Sistem Pengukuran Aliran*. Surabaya : Departemen Teknik Instrumentasi, FV-ITS
- [5] Nurcahyo. 2012. Aplikasi dan Teknik Pemrograman Mikrokontroler AVR Atmel CV Andi Offset.
- [6] Dr. Sri Purnomo Sari, S. M. (n.d.). PENGARUH JARAK DAN UKURAN NOZZLE.
- [7] Texas Instrument. 2014.Hall-Effect Sensor with PNP or NPN Output Manual Book.
- [8] Politeknik Negeri Elektronika Surabaya.
bima.lecturer.pens.ac.id/materi%20praktikum%20dsp/ps2_c odec_sampling.pdf. n.d. bima.lecturer.pens.ac.id (accessed January 29, 2018).

LAMPIRAN

[1] Ultrasonic Ranging Module HC - SR04

□ Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact

measurement function, the ranging accuracy can reach to 3mm. The modules

includes ultrasonic transmitters, receiver and control circuit. The basic principle

of work:

(1) Using IO trigger for at least 10us high level signal,

(2) The Module automatically sends eight 40 kHz and detect whether there is a

pulse signal back.

(3) IF the signal back, through high level , time of high output IO duration is

the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

□ Wire connecting direct as following:

5V Supply

Trigger Pulse Input

Echo Pulse Output

0V Ground

Electric Parameter

Working Voltage DC 5 V

Working Current 15mA

Working Frequency 40Hz

Max Range 4m

Min Range 2cm

MeasuringAngle 15 degree

Trigger Input Signal 10uS TTL pulse

Echo Output Signal Input TTL lever signal and the range in

proportion

Dimension 45*20*15mm

Vcc Trig Echo GND

Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS

pulse to the trigger input to start the ranging, and then the module will send out

an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a

distance object that is pulse width and the range in proportion .You can

calculate the range through the time interval between sending trigger signal and

receiving echo signal. Formula: $\mu\text{S} / 58 = \text{centimeters}$ or $\mu\text{S} / 148 = \text{inch}$; or: the

range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms

measurement cycle, in order to prevent trigger signal to the echo signal.

□ Attention:

The module is not suggested to connect directly to electric, if connected

electric, the GND terminal should be connected the module first, otherwise,

it will affect the normal work of the module.

When tested objects, the range of area is not less than 0.5 square meters

and the plane requests as smooth as possible, otherwise ,it will affect the

results of measuring.

[2] *Datasheet Water Flow Sensor*

Introduction Water flow sensor consists of a plastic valve body, a water rotor, and a hall-effect sensor. When water flows through the rotor, rotor rolls. Its speed changes with different rate of flow. The hall-effect sensor outputs the corresponding pulse Signal. Model:POW110D3B

Program type : Application
AVR Core Clock frequency: 16.000000 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 256

/

```
#include <mega16.h>
```

```
// Alphanumeric LCD Module functions
```

```
#include <alcd.h>
```

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

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

```
#define stepB PORTA.6
```

```
#define stepD PORTA.4
```

```
#define stepC PORTA.5
```

```
#define stepA PORTA.7
```

```
#define Valve_safety PORTA.0
```

```
#define Valve_out PORTA.1
```

```
#define pump PORTA.2
```

```
#define pump2    PORTA.3
```

```
#define ledrun   PORTD.7
```

```
#define buzzer   PORTD.6
```

```
#define ultrasound PORTC.1
```

```
#define maxstep  1000
```

```
int pulse,minperiode,periode,respon,direct,stepcount,level;
```

```
bit refresh,stepper;
```

```
unsigned int range;
```

```
// External Interrupt 0 service routine
```

```
interrupt [EXT_INT0] void ext_int0_isr(void)
```

```
{
```

```
// Place your code here
```

```
if (PIND.2==1)
```

```
{
```

```
    TCCR2=0x02;
```

```
    TCNT2=0x00;
```

```
    range=0;
```

```

    }
    else
    {
        TCCR2=0x00;
        TCNT2=0x00;
        level=48-((range+TCNT2)/116);
        if (level<=0)level=0;
    }
}

// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
    // Place your code here

    periode=pulse;
    minperiode=TCNT0;
    TCNT0=0;
    pulse=0;
}

#ifndef RXB8
#define RXB8 1

```

#endif

#ifndef TXB8

#define TXB8 0

#endif

#ifndef UPE

#define UPE 2

#endif

#ifndef DOR

#define DOR 3

#endif

#ifndef FE

#define FE 4

#endif

#ifndef UDRE

#define UDRE 5

#endif

#ifndef RXC

```
#define RXC 7

#endif

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// USART Receiver buffer
#define RX_BUFFER_SIZE 4
char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE <= 256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
unsigned int rx_wr_index,rx_rd_index,rx_counter;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
```

```

interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
if((data=='#')||(data=='*'))          // Inisial stx (0x23 = #)
{
rx_counter=0;          // rx_counter mulai dari 0
rx_wr_index=0;        // rx_wr_index mulai dari 0
}
if ((status & (FRAMING_ERROR | PARITY_ERROR |
DATA_OVERRUN))==0)
{
rx_buffer[rx_wr_index++]=data;
#if RX_BUFFER_SIZE == 256
// special case for receiver buffer size=256
if (++rx_counter == 0)
{
#else
if (rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
if (++rx_counter == RX_BUFFER_SIZE)
{
rx_counter=0;

```

```

#endif

    rx_buffer_overflow=1;
}
}
}

#ifdef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
    char data;

    while (rx_counter==0);
    data=rx_buffer[rx_rd_index++];
    #if RX_BUFFER_SIZE != 256
    if (rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
    #endif
    #asm("cli")
    --rx_counter;
    #asm("sei")
    return data;
}

```

```
#pragma used-
```

```
#endif
```

```
// Standard Input/Output functions
```

```
#include <stdio.h>
```

```
// Timer 0 overflow interrupt service routine
```

```
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
```

```
{
```

```
// Place your code here
```

```
    pulse++;
```

```
}
```

```
// Timer1 overflow interrupt service routine
```

```
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
```

```
{
```

```
// Place your code here
```

```
    respon++;
```

```
    if (respon>=122)
```

```
    {
```

```
        refresh=1;
```

```
        respon=0;
```

```
    }
```

```

    stepper=1;
}

// Timer2 overflow interrupt service routine
interrupt [TIM2_OVF] void timer2_ovf_isr(void)
{
// Place your code here
    range=range+256;
    if (range>=9280)
    {
        TCCR2=0x00;
        level=48;}
}

moving_stepper(int i)
{
    if (direct==1) //valve buka
    { switch (i)
    {
        case 0: {stepA=0; stepB=0; stepC=0; stepD=0; break;}
        case 1: {stepA=0; stepB=0; stepC=0; stepD=1; break;}
        case 2: {stepA=0; stepB=0; stepC=0; stepD=1; break;}
        case 3: {stepA=0; stepB=1; stepC=0; stepD=0; break;}
    }
}

```

```

        case 4: {stepA=0; stepB=1; stepC=0; stepD=0; break;}
        case 5: {stepA=0; stepB=0; stepC=1; stepD=0; break;}
        case 6: {stepA=0; stepB=0; stepC=1; stepD=0; break;}
        case 7: {stepA=1; stepB=0; stepC=0; stepD=0; break;}
        case 8: {stepA=1; stepB=0; stepC=0; stepD=0; break;}
    }
    stepcount++;if
(stepcount>=maxstep){direct=0;stepcount=maxstep;stepA=0;
stepB=0; stepC=0; stepD=0;}
}
else if(direct==2) //valve tutup
{
switch (i)
{

        case 0: {stepA=0; stepB=0; stepC=0; stepD=0; break;}
        case 1: {stepA=1; stepB=0; stepC=0; stepD=0; break;}
        case 2: {stepA=1; stepB=0; stepC=0; stepD=0; break;}
        case 3: {stepA=0; stepB=0; stepC=1; stepD=0; break;}
        case 4: {stepA=0; stepB=0; stepC=1; stepD=0; break;}
        case 5: {stepA=0; stepB=1; stepC=0; stepD=0; break;}
        case 6: {stepA=0; stepB=1; stepC=0; stepD=0; break;}
        case 7: {stepA=0; stepB=0; stepC=0; stepD=1; break;}
        case 8: {stepA=0; stepB=0; stepC=0; stepD=1; break;}

```

```

    }
;if(stepcount<=0){ stepcount=0;direct=0;stepA=0;      stepcount--
stepC=0; stepD=0;}      stepB=0;
    }

    else if (direct==0){stepA=0; stepB=0; stepC=0; stepD=0;}
//valve mati

}

// Declare your global variables here

void main(void)
{
// Declare your local variables here
    int valve_open,frek,p,flow,setpoint;
    unsigned char temp[16], skip[3];
// Input/Output Ports initialization
// Port A initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out
Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0
State1=0 State0=0
PORTA=0x0F;
DDRA=0xFF;

```

```
// Port B initialization
```

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

```
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0  
State1=0 State0=0
```

```
PORTB=0x00;
```

```
DDRB=0xFF;
```

```
// Port C initialization
```

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

```
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0  
State1=0 State0=0
```

```
PORTC=0x00;
```

```
DDRC=0xFF;
```

```
// Port D initialization
```

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

```
// State7=0 State6=0 State5=0 State4=0 State3=P State2=T  
State1=T State0=T
```

```
PORTD=0x08;
```

```
DDRD=0xF0;
```

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: 62.500 kHz

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0=0x02;

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: 250.000 kHz

// Mode: Fast PWM top=OCR1A

// OC1A output: Discon.

// OC1B output: Discon.

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: On

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=0x03;

TCCR1B=0x1B;

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x04;

OCR1AL=0xE2;

OCR1BH=0x00;

OCR1BL=0x00;

// Timer/Counter 2 initialization

// Clock source: System Clock

// Clock value: 2000.000 kHz

// Mode: Normal top=0xFF

// OC2 output: Disconnected

ASSR=0x00;

TCCR2=0x02;

TCNT2=0x00;

OCR2=0x00;

// External Interrupt(s) initialization

// INT0: On

```
// INT0 Mode: Any change
// INT1: On
// INT1 Mode: Falling Edge
// INT2: Off
GICR|=0xC0;
MCUCR=0x09;
MCUCSR=0x00;
GIFR=0xC0;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x45;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x67;
```

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

// ADC initialization
// ADC disabled
ADCSRA=0x00;

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

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

// Alphanumeric LCD initialization
// Connections specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD
menu:
```

```
// RS - PORTB Bit 0
// RD - PORTB Bit 1
// EN - PORTB Bit 2
// D4 - PORTA Bit 4
// D5 - PORTA Bit 5
// D6 - PORTA Bit 6
// D7 - PORTA Bit 7
// Characters/line: 20
lcd_init(20);

// Global enable interrupts
#asm("sei")
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Simulator Level");
lcd_gotoxy(0,1);
lcd_putsf("Naimah & Nafik");

level=0;
stepcount=maxstep;direct=2;//tutup valve
p=0;
while(stepcount>0)
{ if (stepper==1)
```

```

        { p++;
          if (p>=9) p=1;
          moving_stepper(p);stepper=0;
        }
};

ledrun=1;
direct=0;
lcd_clear();
setpoint=0;
pump2=0;
pump=1;
flow=0;
while (1)
    {   if(rx_buffer_overflow==1)           //penanda
        flag awal data
        {
            if
            ((rx_buffer[0]=='*')&&(rx_buffer[1]=='0')&&(rx_buffer[2]=='0')
            &&(rx_buffer[3]=='0'))pump=1;
            else
            if
            ((rx_buffer[0]=='#')&&(rx_buffer[1]=='1')&&(rx_buffer[2]=='1')
            &&(rx_buffer[3]=='1'))pump=0;
            else if (rx_buffer[0]=='@')
            {
                skip[0]=rx_buffer[1];

```

```
        skip[1]=rx_buffer[2];
        skip[2]=rx_buffer[3];
        setpoint=atoi(skip);

    }
    rx_buffer_overflow=0;
}
if (stepper==1)
{
    p++;
    if (p>=9)
    {
        p=1;

        // printf("%c%4d%c", '$',flow,'#');
    }

    moving_stepper(p);
    stepper=0;

}
}
```

```
// Place your code here
```

```
if (refresh==1)
{
```

```

//printf("%c%4d%c",'@',level,'#');
printf ("%d;", level);
lcd_gotoxy(0,0);
lcd_putsf("Level = ");
sprintf(temp,"%2d cm    ",level);
lcd_puts(temp);

if((periode==0)&&(minperiode==0))frek=0; //flow
frek=1/((periode*0.000128)+(minperiode*0.0000005));

flow=frek*0.13;//dikali dengan hasil kalibrasi

lcd_gotoxy(0,1);
lcd_putsf("Flow =");//tampilan lcd
sprintf(temp,"%4d flow",flow);
lcd_puts(temp);
printf("%d", flow);

valve_open=stepcount/(maxstep/100);
lcd_gotoxy(0,2);
lcd_putsf("Valve = ");
sprintf(temp,"%3d %c",valve_open,'%');

```

```
printf("%d;\r\n",valve_open);
```

```
lcd_puts(temp);
```

```
ultrasound=0;
```

```
ultrasound=1;
```

```
delay_us(10);
```

```
ultrasound=0;
```

```
ledrun=1;
```

```
    //progam penegndalian motor stepper
```

```
    if (setpoint>=valve_open+2)direct=1;
```

```
    else if (setpoint<=valve_open-2)direct=2;
```

```
    else if
```

```
((setpoint<valve_open+2)&&(setpoint>valve_open-2))direct=0;
```

```
    //progam setpoint valve
```

```
    if (level>=20+1){setpoint=0;}
```

```
    else if (level<=20-1)setpoint=100;
```

```
    if (level>=20+5)Valve_safety=1;
```

```
    else Valve_safety=0;
```

```
    if (level>=10)Valve_out=1;
```

```
        refresh=0;
    }
}
}
```

y 29, 2018).

BIODATA



Nafi' Kurniawan yang biasa dipanggil "Nafi" ini merupakan mahasiswa dari daerah Mojokerto. Penulis menempuh pendidikan formal di SDN Kranggan 5, SMP N 4 Mojokerto, dan SMKN 1 Mojokerto. Pada tahun 2015 penulis mengikuti test ujian D3 ITS dan diterima pada jurusan D3 Metrologi dan Instrumentasi ini. Apabila ada pertanyaan mengenai tugas akhir penulis dapat menghubungi 083832028206 atau melalui email nafiskanesa@gmail.com