

# **METODA PENANGGULANGAN BANJIR DI LABORATORIUM OTOMASI**

## **TUGAS AKHIR**

*Diajukan Untuk Memenuhi Salah Satu Syarat Kelulusan Dalam Mengikuti  
Program Sarjana Strata-1*

Disusun Oleh :

**DODI HIDAYAT**

**083030061**



**JURUSAN TEKNIK MESIN  
FAKULTAS TEKNIK  
UNIVERSITAS PASUNDAN  
BANDUNG  
2015**

# METODA PENANGGULANGAN BANJIR DI LABORATORIUM OTOMASI

---

## TUGAS AKHIR



**Nama : Dodi Hidayat**

**NRP : 083030061**

**Pembimbing I**

**Pembimbing II**

**Rachmad Hartono, Ir., MT.**  
**Pembimbing Tugas Akhir**

**Sugiharto, Ir., MT.**  
**Pembimbing Tugas Akhir**

## **ABSTRAK**

Genangan air yang terjadi di laboratorium otomasi terjadi saat hujan. Air hujan dari luar merembes lewat dinding laboratorium yang mengakibatkan sarana dan prasarana yang berada di laboratorium tergenang air dan rusak.

Sistem penanggulangan banjir meliputi berbagai cara dan sarana, yang salah satunya adalah penampungan air. Penampungan air yang ada saat ini mempunyai kelemahan yaitu air meluap tanpa diketahui. Salah satu alternatif untuk mengatasi hal ini adalah dengan membuat alat pendeteksi ketinggian air di penampungan air berbasis mikrokontroler ATmega8535. Pendeteksi ini menggunakan sensor ultrasonik ping parallax.

Sensor ultrasonik adalah alat elektronika yang mengubah energi listrik menjadi energi mekanik dalam bentuk gelombang [suara](#) ultrasonik. Sensor ultrasonik berfungsi untuk mendeteksi perubahan ketinggian air yang memanfaatkan gelombang ultrasonik yang dipantulkan kemudian menghitung jarak gelombang yang terpantulkan kembali ketika menyentuh permukaan air di penampungan. Gelombang yang diserap akan dihitung oleh komparator dan diteruskan menjadi bilangan binary kemudian mikrokontroler mendata bilangan sesuai dengan penginputan untuk menghidupkan pompa air

## KATA PENGANTAR



### **Assalamu'alaikum, Wr.Wb.**

Syukur alhamdulillah penulis panjatkan puji dan syukur kehadirat Allah SWT Yang Maha Pengasih dan Maha Penyayang yang telah melimpahkan hidayah, inayah, dan rahmat-Nya sehingga Laporan Tugas Akhir ini yang diberi judul **“METODA PENANGGULANGAN BANJIR DI LABORATORIUM OTOMASI”** dapat diselesaikan dengan baik.

Laporan Tugas Akhir ini diajukan untuk memenuhi salah satu syarat kelulusan akademik dalam mengikuti program Sarjana Strata-1 (S1) di Fakultas Teknik Jurusan Teknik Mesin Universitas Pasundan Bandung. Walaupun demikian, penulis menyadari sepenuhnya akan kekurangan yang terdapat pada penulisan Laporan Tugas Akhir ini.

Meskipun banyak kendala dan rintangan dalam menyelesaikan laporan ini, tetapi berkat bantuanyang diperoleh penulis dari banyak pihak maka penulis dapat menyelesaikan laporan ini. Oleh karena itu dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih yang sedalam – dalamnya kepada:

1. Ayahanda dan Ibunda tercinta, kakak beserta keluarga besar penulis, terima kasih atas do'a dan segala pengorbanan yang telah diberikan selama ini. Semoga Allah SWT Yang Maha Pengasih dan Maha Penyayang membalas dengan balasan yang sebaik - baiknya dan menyayangi mereka seperti mereka yang menyayangi penulis. Amin.
2. Bapak Rachmad Hartono, Ir.MT., selaku Dosen Pembimbing I. Terimakasih atas ilmu yang begitu berguna, motivasi yang tak pernah henti, filosofi hidup yang telah diajarkan baik yang tersirat maupun tersurat dan kesabaran serta keikhlasan dalam membimbing penulis.
3. Bapak Sugiharto, Ir.,MT., selaku Dosen Pembimbing II. Terimakasih atas semua ilmu yang telah diberikan, perhatian, kesabaran dan inspirasinya.

4. Seluruh Dosen Jurusan Teknik Mesin Universitas Pasundan yang telah memberikan ilmunya selama ini.
5. Siti Latifah, terima kasih atas perhatian, dukungan, motivasi dan semangat yang takhenti – hentinya diberikan selama penulis mengenalnya.
6. Rekan-rekan seperjuangan di laboratorium **PEOTRO (Pesantren Otomasi dan Robotika)** yang telah meluangkan waktunya untuk membantu dan member dukungan kepada penulis selama ini.  
“kapan kita gebleg – geblegan lagi brow??”.
7. Seluruh rekan – rekan teknik mesin kelas reguler dan kelas non reguler (karyawan) yang tidak bisa penulis sebutkan satu per satu terima kasih atas dukungan, motivasi dan do’anya.

Laporan tugas akhir ini masih banyak kekurangan – kekurangan yang harus diperbaiki. Hal ini semata – mata keterbatasan penulis dalam menyusun laporan tugas akhir ini. Oleh karena itu, saran dan kritik yang bersifat membangun sangat diharapkan demi kesempurnaan laporan Tugas Akhir ini. Semoga Laporan Tugas Akhir ini bermanfaat khususnya bagi penulis dan umumnya bagi pembaca.

Akhir kata dengan segala kerendahan hati, penulis memohon kepada Allah SWT agar dapat membalas segala kebaikan bagi mereka yang telah membantu, **Amin...**

**Wassalamu’alaikum. Wr, Wb.**

Bandung, 19 Januari 2015

**Dodi Hidayat**

Penulis

## DAFTAR ISI

### ABSTRAK

KATA PENGANTAR .....	i
DAFTAR ISI .....	iii
DAFTAR GAMBAR .....	v
DAFTAR TABEL .....	vii

### BAB I PENDAHULUAN

1.1 Latar Belakang Masalah .....	I-1
1.2 Batasan Masalah .....	I-2
1.3 Tujuan .....	I-2
1.4 Skematik Penulisan .....	I-2

### BAB II TEORI DASAR

2.1 Penanggulangan Banjir Di Laboratorium Otomasi .....	II-1
2.2 Sensor Ultrasonic Ping Parallax .....	II-1
2.3 Mikrokontroler ATmega8535 .....	II-2
2.4 IC L293D .....	II-3
2.5 LCD ( <i>Liquid Crystal Display</i> ) .....	II-4
2.6 Pompa Air .....	II-5
2.7 <i>CodeVision AVR</i> .....	II-6

### BAB III PEMBUATAN RANGKAIAN DAN PERANGKAT LUNAK UNTUK MENANGGULANGI BANJIR DI LABORATORIUM OTOMASI

3.1 Rangkaian Kontrol Metoda Penanggulangan Banjir Di Laboratorium Otomasi .....	III-1
3.2 Rangkaian Sensor Ultrasonik .....	III-2
3.3 Rangkaian <i>Liquid Crystal Display</i> (LCD) .....	III-4
3.4 <i>Driver</i> Pompa Air .....	III-5

### BAB IV PENGUJIAN DAN ANALISA

4.1 Pengujian Sensor Ultrasonik Ping Parallax .....	IV-1
---	------

## ***Laporan Tugas Akhir***

---

4.2 Pengujian <i>Driver</i> Pompa Air .....	IV-4
4.3 Pengujian Alat Keseluruhan .....	IV-5
4.4 Analisa Hasil Pengujian .....	IV-6

## **BAB V KESIMPULAN DAN SARAN**

5.1 Kesimpulan .....	V-1
5.2 Saran .....	V-1

## **DAFTAR PUSTAKA**

## **LAMPIRAN**

**DAFTAR GAMBAR**

Gambar 2.1	Sensor Ultrasonik Ping Parallax .....	II-2
Gambar 2.2	Bentuk IC ATmega8535 .....	II-3
Gambar 2.3	Bentuk dan Simbol IC L293D .....	II-3
Gambar 2.4	LCD 16*2 .....	II-5
Gambar 2.5	Pompa Air .....	II-6
Gambar 2.6	Tampilan Awal Program <i>CodeVisionAVR</i> .....	II-7
Gambar 2.7	Konfirmasi penggunaan <i>CodeWizardAVR</i> .....	II-7
Gambar 2.8	Konfigurasi Program Melalui <i>CodeWizardAVR</i> .....	II-8
Gambar 2.9	<i>Form Program</i> .....	II-9
Gambar 2.10	Kotak Dialog Information Configure .....	II-9
Gambar 3.1	Skema Diagram Blok Rangkaian Penanggulangan Banjir Di Laboratorium Otomasi .....	III-1
Gambar 3.2	Sensor Ultrasonik Parallax Sebagai Pendeteksi Ketinggian Air Di Penampungan .....	III-2
Gambar 3.3	Diagram Waktu Sensor Ultrasonik Ping Parallax .....	III-3
Gambar 3.4	Rangkaian LCD 16*2 .....	III-5
Gambar 3.5	Rangkaian <i>Driver</i> Pompa Air .....	III-6
Gambar 3.6	Skematik Rangkaian <i>Driver</i> Pompa Air .....	III-6
Gambar 4.1	Kurva Penunjukan LCD vs Jarak Sebenarnya .....	IV-3
Gambar 4.2	Keakuratan Penunjukan LCD Terhadap Jarak Sebenarnya .....	IV-4
Gambar 4.3	Skematik <i>Driver</i> Pompa Air .....	IV-4
Gambar 4.4	Rangkaian Keseluruhan	



***Laporan Tugas Akhir***

---

Alat Penanggulangan Banjir .....	IV-5
Gambar 4.5 Penunjukan Angka 360 mm Pada LCD .....	IV-6
Gambar 4.6 Penunjukan Angka 385 mm PadaLCD .....	IV-6

**DAFTAR TABEL**

Tabel 2.1	Susunan Kaki LCD 16*2 .....	II-4
Tabel 2.2	Spesifikasi Pompa .....	II-5
Tabel 3.1	Program Pembacaan Sensor Ultrasonik .....	III-4
Tabel 3.2	Fungsi Menuliskan Angka Pada LCD .....	III-5
Tabel 4.1	Program Pengujian Sensor Ultrasonik .....	IV-1
Tabel 4.2	Hasil Pengujian Sensor Ultrasonik .....	IV-2
Tabel 4.3	List Program Dengan Menggunakan Persamaan .....	IV-3

## **BAB I PENDAHULUAN**

Pada bab ini akan dibahas mengenai latar belakang, batasan masalah, tujuan, dan sistematika penulisan.

### **1.1. Latar Belakang Masalah**

Genangan air di laboratorium otomasi robotika terjadi saat hujan. Air hujan dari luar merembes lewat dinding laboratorium yang mengakibatkan sarana dan prasarana yang berada di laboratorium tergenang air dan rusak. Pembuatan penampungan air sudah dilakukan namun terdapat masalah yaitu air meluap tanpa diketahui. Oleh karena itu dibutuhkan suatu metoda yang dapat menangani luapan air dari penampungan tersebut secara efisien. Untuk menangani luapan air tersebut diantaranya dapat dilakukan dengan cara mengalirkan air dari penampungan keluar ruangan dengan menggunakan pompa yang dapat bekerja secara otomatis.

Salah satu cara untuk melengkapi kerja pompa secara otomatis yaitu dengan merancang perangkat sistem pendeteksi ketinggian permukaan air di penampungan air berbasis mikrokontroler ATMega8535. Pendeteksi ini menggunakan sensor ping parallax untuk mendeteksi perubahan jarak ketinggian air yang memanfaatkan gelombang ultrasonik.

Sensor ultrasonik adalah alat elektronika yang mengubah energi listrik menjadi energi mekanik dalam bentuk gelombang suara ultrasonik. Gelombang Ultrasonik dapat merambat melalui zat padat, cair maupun gas. Prinsip kerja sensor ultrasonik yaitu sinyal dipancarkan melalui pemancar gelombang ultrasonik, sinyal yang dipancarkan akan merambat sebagai gelombang bunyi dengan kecepatan bunyi berkisar 344 m/s. Sinyal yang sudah diterima akan diproses untuk menghitung jaraknya.

Penanggulangan banjir di laboratorium otomasi merupakan suatu rangkaian pencegahan terjadinya banjir. Dimana alat tersebut memanfaatkan gelombang ultrasonik sebagai pendeteksi ketinggian air. Gelombang ultrasonik yang dipantulkan kemudian menghitung jarak

gelombang yang terpantulkan kembali ketika menyentuh permukaan air. Perubahan jarak ketinggian air akan ditampilkan melalui LCD (*Liquid Crystal Display*), Kemudian data diolah oleh mikrokontroler ATmega8535 dan hasilnya dikirim untuk menyalakan pompa air sesuai dengan penginputan data pada mikrokontroler ATmega8535.

## **1.2. Batasan Masalah**

Masalah yang dibahas dalam tugas akhir ini meliputi pembuatan sistem kontrol pompa air secara otomatis dengan menggunakan rangkaian mikrokontroler dan sensor ultrasonik.

## **1.3. Tujuan**

Tujuan dari tugas akhir ini adalah merancang dan membuat alat penanggulangan banjir di laboratorium otomasi.

## **1.4. Sistematika Penulisan**

Laporan tugas akhir ini terdiri dari lima bab. Isi masing-masing bab adalah sebagai berikut:

### **BAB I PENDAHULUAN**

Bab ini berisi tentang latar belakang masalah, batasan masalah, tujuan, dan sistematika penulisan.

### **BAB II TEORI DASAR**

Bab ini berisi tentang teori-teori yang mendukung dan berkaitan langsung dengan proses perancangan dan pembuatan alat penanggulangan banjir di laboratorium otomasi.

### **BAB III PEMBUATAN RANGKAIAN DAN PERANGKAT LUNAK UNTUK MENANGGULANGI BANJIR DI LABORATORIUM OTOMASI**

Pada bab ini dibahas tentang, rangkaian kontrol penanggulangan banjir di laboratorium otomasi dan pembuatan program penanggulangan banjir di laboratorium otomasi

### **BAB IV PENGUJIAN DAN ANALISA**

Bab ini berisi tentang pengujian dan analisa alat penanggulangan banjir di laboratorium otomasi.

### **BAB V KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan dan saran yang berhubungan dengan alat penanggulangan banjir di laboratorium otomasi.

### **DAFTAR PUSTAKA**

### **LAMPIRAN**

## **BAB II TEORI DASAR**

Pada bab ini dibahas tentang penanggulangan banjir, sensor ultrasonik ping parallax, mikrokontroler ATmega8535, IC L293D, LCD, pompa air dan *CodeVisionAVR*.

### **2.1. Penanggulangan Banjir di Laboratorium Otomasi**

Genangan air yang terjadi di laboratorium otomasi terjadi saat hujan. Air hujan dari luar merembes lewat dinding laboratorium yang mengakibatkan sarana dan prasarana yang berada di laboratorium tergenang air dan rusak. Sistem penanggulangan banjir meliputi berbagai cara dan sarana, yang salah satunya adalah penampungan air. Penampungan air yang ada saat ini mempunyai kelemahan yaitu air meluap tanpa diketahui. Salah satu alternatif untuk mengatasi hal ini adalah dengan membuat alat pendeteksi ketinggian air di penampungan air berbasis mikrokontroler ATmega8535. Pendeteksi ini menggunakan sensor ping parallax untuk mendeteksi perubahan ketinggian air yang memanfaatkan gelombang ultrasonik yang dipantulkan kemudian menghitung jarak gelombang yang terpantulkan kembali ketika menyentuh permukaan air di penampungan. Gelombang yang diserap akan dihitung oleh komparator dan diteruskan menjadi bilangan binary kemudian mikrokontroler mendata bilangan sesuai dengan penginputan untuk menghidupkan pompa air.

### **2.2. Sensor Ultrasonik Ping Parallax**

Sensor ultrasonik ping parallax adalah alat elektronik yang mempunyai kemampuan mengubah energi listrik menjadi energi mekanik dalam bentuk gelombang suara ultrasonik. Sensor ini terdiri dari rangkaian pemancar ultrasonik yang dinamakan *transmitter* dan penerima ultrasonik yang disebut *receiver*.

Gelombang ultrasonik adalah gelombang dengan besar frekuensi diatas frekuensi gelombang suara yaitu lebih dari 20 KHz. Sensor Ultrasonik Ping parallax ini dapat mengukur jarak antara 3 cm sampai 300 cm. Keluaran sensor ini mempresentasikan jarak lama waktu sejak sinyal ultrasonik dipancarkan oleh sensor sampai sinyal tersebut diterima kembali. Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari *transmitter* ultrasonik. Ketika sinyal mengenai benda penghalang, sinyal ini dipantulkan, dan diterima oleh *receiver* ultrasonik. Sinyal yang diterima oleh rangkaian *receiver* dikirimkan ke rangkaian mikrokontroler, dan selanjutnya diolah untuk menghitung jarak terhadap benda di depannya (bidang pantul). Bentuk sensor ultrasonik ping parallax dapat dilihat pada gambar 2.1.



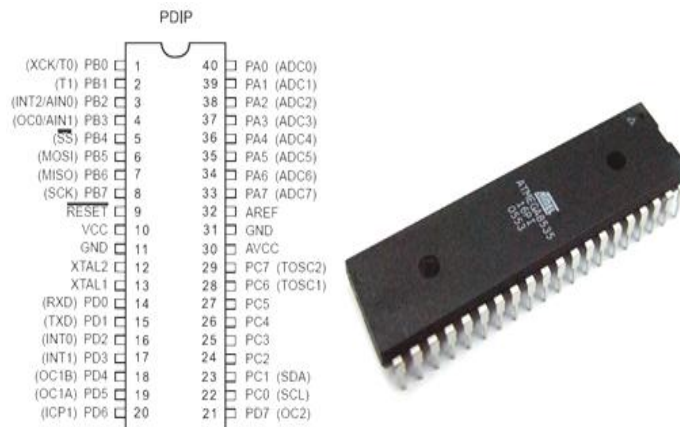
**Gambar 2.1 Sensor Ultrasonik Ping Parallax**

### **2.3. Mikrokontroler ATmega8535**

Mikrokontroler merupakan perangkat elektronika yang didalamnya terdapat rangkaian kontrol, mikroprosesor, memori, dan *input/output*. Mikrokontroler dapat diprogram menggunakan berbagai macam bahasa pemrograman. Bahasa pemrograman yang biasa digunakan untuk memprogram mikrokontroler diantaranya adalah bahasa *assembler*, bahasa C, bahasa *basic* dan lain-lain.

Mikrokontroler biasanya digunakan untuk mengendalikan suatu proses secara otomatis seperti sistem kontrol mesin, *remote* kontrol, kontrol alat berat, kontrol robot dan lain-lain. Dengan menggunakan mikrokontroler sistem kontrol akan menjadi lebih ringkas, lebih mudah, dan lebih ekonomis.

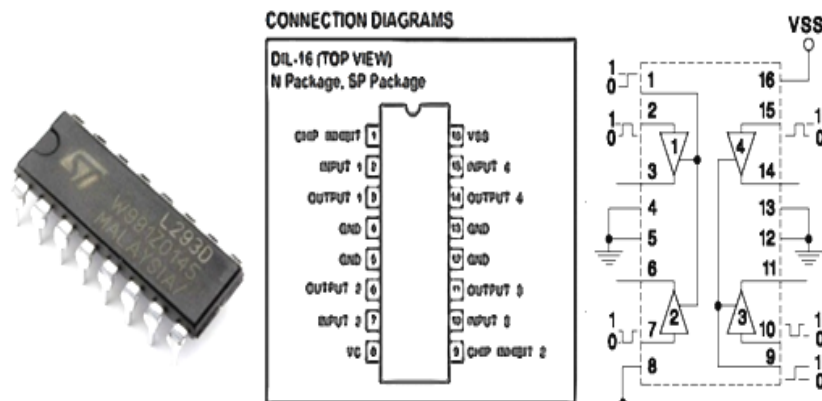
Salah satu jenis mikrokontroler yang banyak digunakan untuk aplikasi kontrol adalah ATmega8535. ATmega8535 merupakan salah satu mikrokontroler keluaran Atmel. Atmel adalah salah satu vendor yang bergerak dibidang mikroelektrika. ATmega8535 memiliki beberapa fitur yang dapat digunakan untuk aplikasi kontrol. Bentuk mikrokontroler ATmega8535 dapat dilihat pada gambar 2.2.



**Gambar 2.2 Bentuk IC ATmega8535**

#### 2.4. IC L293D

IC L293D adalah komponen elektronika yang dapat digunakan untuk mengontrol sebuah motor. Selain itu L293D mampu menjalankan beban induktif seperti *relay*, katup *solenoid*, motor DC maupun *motor stepper*. Bentuk dan simbol IC L293D dapat dilihat pada gambar 2.3.



**Gambar 2.3 Bentuk dan Simbol IC L293D**



Kaki-kaki *input* terdapat pada kaki 2, 7, 10, dan 15. Kaki-kaki *output* terdapat pada kaki 3, 6, 11 dan 14. Hubungan *input* dan *output* akan berfungsi jika tegangan pada kaki *inhibit* bernilai *high*.

## 2.5. LCD (*Liquid Crystal Display*)

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. LCD sudah digunakan diberbagai bidang misalnya alat-alat elektronik seperti televisi, kalkulator, ataupun layar komputer. LCD yang digunakan pada rancangan metoda penanggulangan banjir adalah tipe LCD dot matrik dengan jumlah karakter 16\*2. LCD sangat berfungsi sebagai penampil yang nantinya akan digunakan untuk menampilkan status kerja alat. Bentuk LCD 16\*2 dapat dilihat pada gambar 2.4. Susunan kaki LCD 16\*2 dapat dilihat pada tabel 2.1.

**Tabel 2.1**  
**Susunan kaki LCD 16\*2**

PIN	Nama Pin	I/O	Keterangan	Hubungan Port Mikrokontroller
1	V <sub>SS</sub>	Power	0V	Ground
2	V <sub>DD</sub>	Power	+5V	Tegangan 5V
3	V <sub>O</sub>	Power	Tegangan kontras LCD	Keluaran trimpot
4	Rs	Input	Register select, 0 = Register perintah, 1= Register data	PA0
5	R/W	Input	Sebagai input, R =1, W = 0, diaktifkan low	
6	E	Input	Enable clock LCD, logika 1 setiap kali pengiriman atau pembacaan data	PA1
7	Db0	I/O	Data bus 0	
8	Db1	I/O	Data bus 1	
9	Db2	I/O	Data bus 2	
10	Db3	I/O	Data bus 3	
11	Db4	I/O	Data bus 4	PA2
12	Db5	I/O	Data bus 5	PA3
13	Db6	I/O	Data bus 6	PA4
14	DB7	I/O	Data bus 7	PA5
15	Anoda	Power	Tegangan positif	
16	Katoda	Power	Tegangan negatif	



**Gambar 2.4 LCD 16\*2**

## **2.6. Pompa Air**

Pompa adalah suatu alat atau mesin yang digunakan untuk memindahkan cairan dari suatu tempat ke tempat yang lain melalui suatu media perpipaan. Cairan tersebut dipindahkan dengan cara menambahkan energi pada cairan yang dipindahkan dan berlangsung secara terus menerus.

Pompa beroperasi dengan prinsip membuat perbedaan tekanan antara bagian masuk (suction) dengan bagian keluar (discharge). Dengan kata lain, pompa berfungsi mengubah tenaga mekanis dari suatu sumber tenaga (penggerak) menjadi tenaga kinetis (kecepatan), tenaga ini berguna untuk mengalirkan cairan dan mengatasi hambatan yang ada di sepanjang pengaliran. Salah satu contoh pompa air dapat dilihat pada gambar 2.5. Spesifikasi pompa dapat dilihat pada table 2.2.

**Tabel 2.2 Spesifikasi Pompa**

Voltage/Hz	220/50 V
Daya Output Motor	125 Watt
Daya Input Motor	350 Watt
Panjang pipa hisap	9 meter
Daya Kerja Keseluruhan max. (Head)	30 meter
Head 5 meter	Kapasitas 25 liter / menit
Head 20 meter	Kapasitas : 10 liter/menit
Pipa Hisap	1 inch
Pipa Dorong	1 inch
Berat	5.5 Kg



**Gambar 2.5 Pompa Air**

## **2.7. CodeVision AVR**

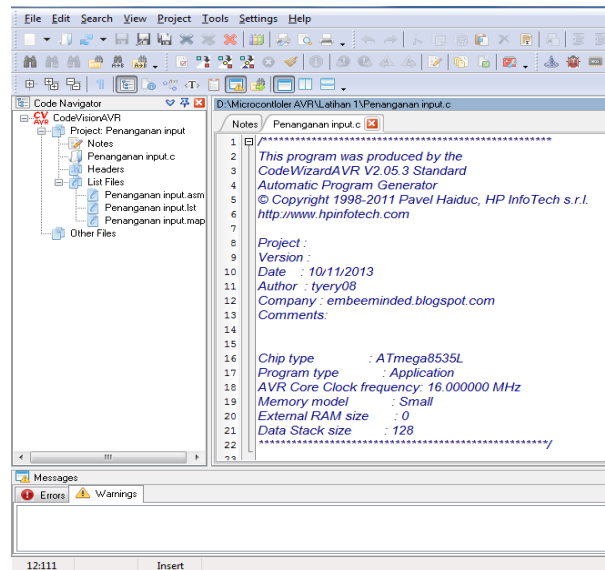
Ada beberapa program yang dapat digunakan sebagai *editor* dan *compiler* untuk mikrokontroler, salah satunya adalah menggunakan program *CodeVisionAVR*. *CodeVisionAVR* adalah salah satu alat bantu pemrograman (*programming tool*) yang bekerja dalam lingkungan pengembangan perangkat lunak yang terintegrasi (*Integrated Development Environment, IDE*). Seperti aplikasi IDE lainnya, *CodeVisionAVR* dilengkapi dengan *source code editor*, *compiler*, *linker*, dan dapat memanggil Atmel AVR Studio untuk *debugger* nya.

Proses *download* program ke IC mikrokontroler AVR dapat dilakukan dengan menggunakan *system download* secara *ISP (In-System Programming)*. *ISP* mengizinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial *SPI*.

Langkah-langkah untuk menjalankan program *CodeVisionAVR* adalah sebagai berikut :

1. Membuka *CodeVisionAVR*

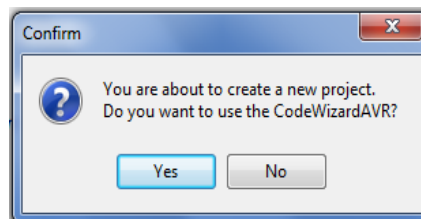
Langkah untuk membuka *CodeVisionAVR* dilakukan dengan memilih menu **Start Menu** → **All Program** → **CodeVisionAVR Evaluation**. Setelah langkah tersebut dilakukan pada monitor akan tampil tampilan seperti pada gambar 2.6.



Gambar 2.6 Tampilan awal program CodeVisionAVR

2. Membuat project baru

Langkah untuk membuat project baru dilakukan dengan memilih menu **File** → **New**, kemudian Pilih dan **Klik Project** . Setelah langkah tersebut dilakukan pada monitor akan muncul kotak dialog *Confirm*. Kotak dialog ini menanyakan apakah CodeWizardAVR akan digunakan atau tidak. Jika *Codewizard* akan digunakan maka tekan tombol YES. Kotak dialog *confirm* dapat dilihat pada gambar 2.7.

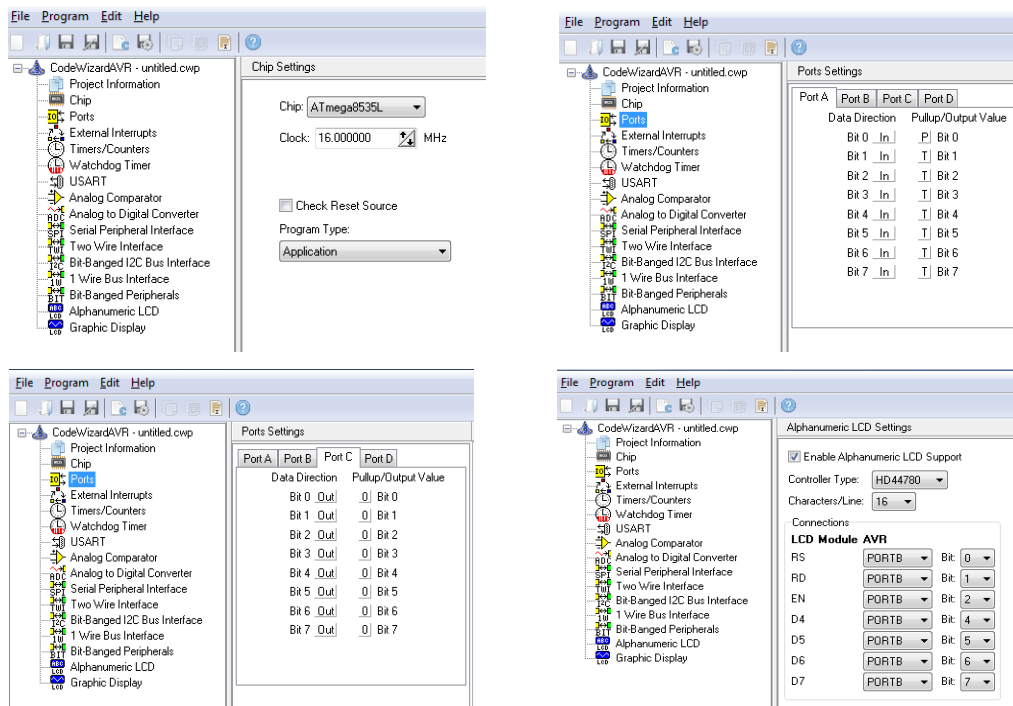


Gambar 2.7 Konfirmasi Penggunaan CodeWizardAVR

3. Melakukan konfigurasi project

Langkah untuk menkonfigurasi project dilakukan dengan memilih tab-tab yang ada pada kotak dialog CodeWizardAVR. Tab-tab yang ada pada kotak dialog CodeWizard terdiri dari tab **USART**, tab **Analog Comperator**, tab **ADC**, tab **SPI**, tab **I<sub>2</sub>C**, tab **1 Wire**, tab **2 Wire (I<sub>2</sub>C)**, tab **LCD**, tab **Bit-Banged**, tab **Project Information**, tab

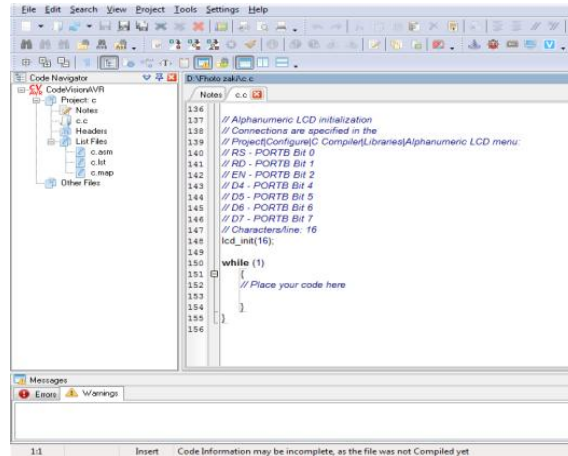
**Chip**, tab **Port**, tab **External IRQ**, dan tab **Timer**. Sebagian tab-tab dapat pada kotak dialog CodeWizardAVR dilihat pada gambar 2.8.



**Gambar 2.8 Konfigurasi Program Melalui CodeWizardAVR**

#### 4. Membuat kode program

Langkah untuk membuat kode program dilakukan dengan membuat file dan menamainya terlebih dahulu, kemudian menyimpannya pada folder yang telah ditentukan. Langkah-langkah pembuatan file dilakukan dengan memilih menu *Program, Generate, save and Exit*, kemudian *file source (\*.c)*, *file project (\*.prj)* dan *file project codewizard (\*.cwp)*, diberi nama dan disimpan pada folder yang telah ditentukan. Setelah langkah tersebut, pada monitor akan tampil program yang siap diisi oleh program yang dibuat. Form program dapat dilihat pada gambar 2.9.




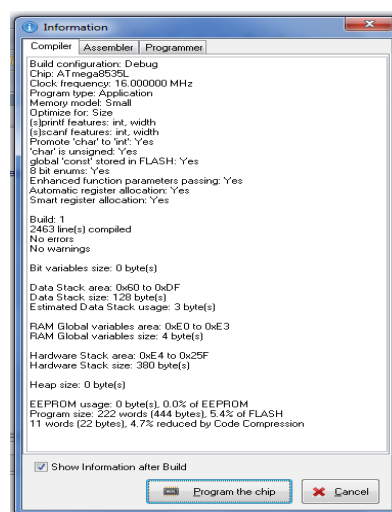
Gambar 2.9 Form Program

5. Meng-configure program

Langkah untuk meng-convigurasi program dilakukan dengan memilih menu **Project** → **Configure** → **AfterBuild** → **program the Chip** → **Ok**.

6. Meng-compile program

Langkah untuk meng-compile program dilakukan dengan memilih menu **Project | compile** atau dengan memilih icon  → Program. Jika penulisan kode program benar, maka akan tampil kotak dialog *information*. Kotak dialog information dapat dilihat pada gambar 2.10.



Gambar 2.10 Kotak Dialog Information Configure

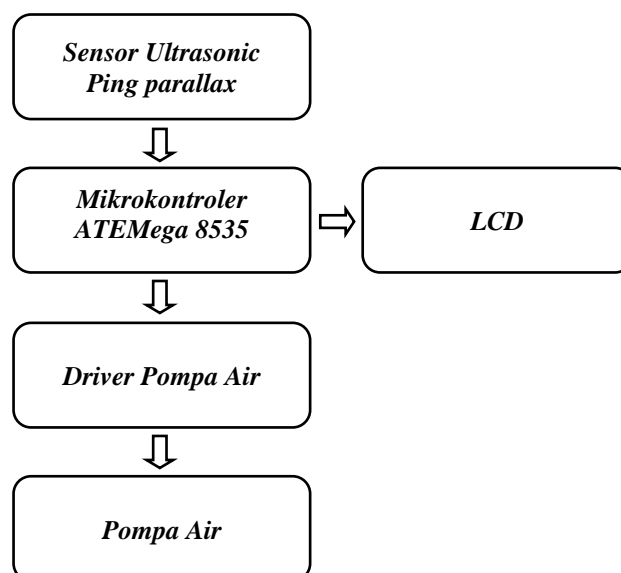
### BAB III

## PEMBUATAN RANGKAIAN DAN PERANGKAT LUNAK UNTUK MENANGGULANGI BANJIR DI LABORATORIUM OTOMASI

Pada bab ini dibahas tentang, rangkaian kontrol penanggulangan banjir di laboratorium otomasi dan pembuatan program penanggulangan banjir di laboratorium otomasi.

### 3.1. Rangkaian Kontrol Metoda Penanggulangan Banjir Di Laboratorium Otomasi

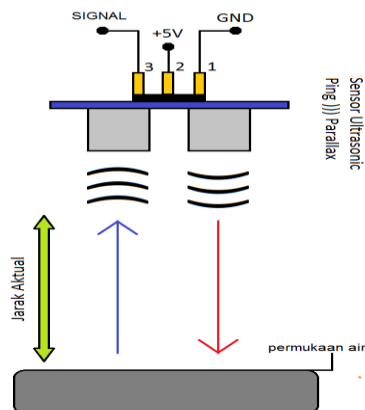
Rangkaian kontrol metoda penanggulangan banjir di laboratorium otomasi merupakan gabungan dari beberapa komponen elektronika sehingga menghasilkan fungsi tertentu. Pembuatan rangkaian metoda penanggulangan banjir di laboratorium otomasi dilakukan dengan cara menghubungkan beberapa rangkaian komponen elektronika. Komponen elektronika yang dihubungkan diantaranya adalah rangkaian sensor ultrasonik ping parallax, mikrokontroler ATmega8535, LCD, *driver* pompa, dan pompa air. Skema diagram blok rangkaian penanggulangan banjir di laboratorium otomasi dapat dilihat pada gambar 3.1.



**Gambar 3.1**  
**Skema Diagram Blok Rangkaian Penanggulangan Banjir**  
**Di Laboratorium Otomasi**

### 3.2. Rangkaian Sensor Ultrasonik

Sensor ultrasonik ping parallax berfungsi sebagai pengukur jarak. Alat tersebut memanfaatkan gelombang ultrasonik sebagai pendeteksi ketinggian air di penampungan. Sensor ini terdiri dari rangkaian pemancar ultrasonik yang dinamakan *transmitter* dan penerima ultrasonik yang disebut *receiver*. Pada rangkain sensor ultrasonik terdapat 3 pin yang digunakan yaitu jalur *power supply* (+5V), ground, dan signal. Pin signal dapat langsung dihubungkan dengan mikrokontroler. Gambar skematik sensor ultrasonik ping parallax sebagai pendeteksi ketinggian air di penampungan dapat dilihat pada gambar 3.2.

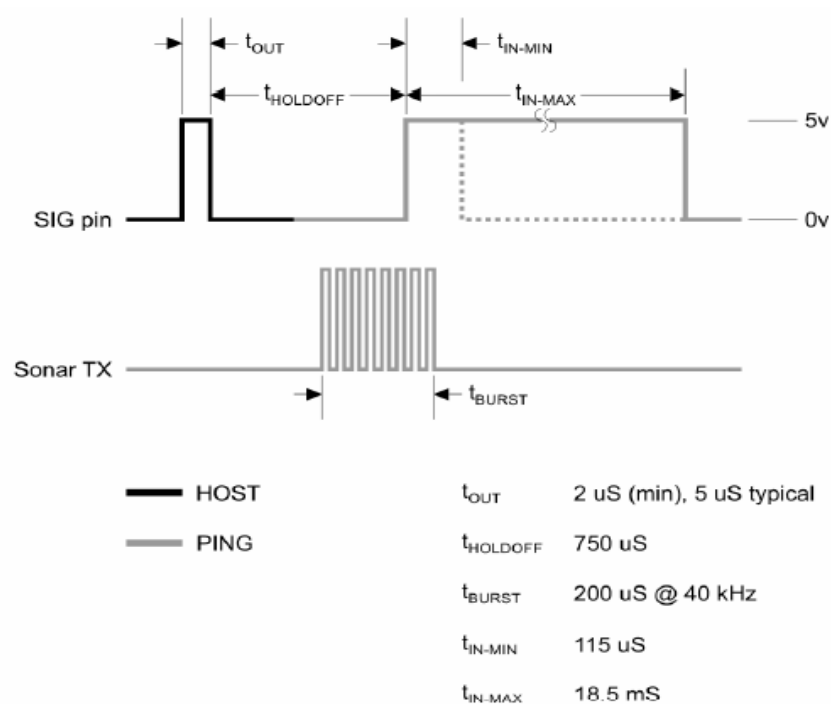


**Gambar 3.2**  
**Sensor Ultrasonik Ping Parallax Sebagai Pendeteksi Ketinggian Air di Penampungan**

Pin signal yang terdapat pada rangkaian sensor ultrasonik dapat berfungsi sebagai *input* maupun *output*. Mikrokontroler memberi pulsa trigger pada sensor ultrasonik sehingga pin signal yang terdapat pada sensor ultrasonik menjadi output. Pengkondisian pin signal menjadi high dilakukan selama 2  $\mu$ s sampai dengan 5  $\mu$ s. Pengkondisian pin signal menjadi high bertujuan untuk mengaktifkan chip pembangkit gelombang ultrasonik. Ketika kondisi pin signal *high* gelombang ultrasonik memancar dari sensor ultrasonik, kemudian kondisi pin signal dijadikan low agar sensor ultrasonik berhenti memancarkan gelombang ultrasonik. Kondisi



pin signal kembali dijadikan *high* dan dijadikan *input* agar pin signal dapat mendeteksi gelombang ultrasonik yang diterima sensor ultrasonik. Ketika menunggu gelombang ultrasonik yang sampai ke sensor, mikrokontroler menambahkan nilai suatu variabel. Nilai variabel yang ditambahkan akan sebanding dengan 2 kali jarak gelombang ultrasonik terhadap benda. Gambar skematik diagram waktu sensor ultrasonik dapat dilihat pada gambar 3.3. Program pada mikrokontroler yang digunakan untuk mengukur jarak antara sensor ultrasonik dengan benda dari awal memberikan *trigger* pulsa *high* sampai didapatkan nilai jarak dari pembacaan sensor dapat dilihat pada tabel 3.2.



**Gambar 3.3**  
**Diagram Waktu Sensor Ultrasonik Ping Parallax**

Ket :

- $t_{OUT}$  : Input Trigger Pulse
- $t_{HOLDOFF}$  : Echo Holdoff
- $t_{BURST}$  : Burst Frequency
- $t_{IN-MIN}$  : Echo Return Pulse Minimum
- $t_{IN-MAX}$  : Echo Return Pulse Maximum

**Tabel 3.1**  
**Program Pembacaan Sensor Ultrasonik**

```
{
// Place your code here
counter=0; //initial value
DirSig=1; //set as output
SigOut=1;
delay_us(5);
SigOut=0;
DirSig=0; //set as input
SigOut=1; //Pullup activated
while (SigIn==0){}
while (SigIn==1)counter++;

distance=(counter*0.109-6.381);

if(distance>=385)PORTC.1=0;
if(distance<=360)PORTC.1=1;

lcd_clear();
lcd_gotoxy(2,0);
lcd_putsf("Jarak");
lcd_gotoxy(4,1);
tulis_angka(distance);
lcd_gotoxy(8,1);
lcd_putsf("mm");
delay_ms(200);
}
```

Setelah program tersebut dituliskan secara lengkap program dapat di-*download* ke mikrokontroler.

### **3.3. Rangkaian *Liquid Crystal Display* (LCD)**

Rangkaian LCD merupakan suatu jenis media tampil yang digunakan sebagai penunjuk pengukuran. Penunjuk hasil pengukuran yang ditampilkan pada LCD adalah data perubahan ketinggian air di penampungan berupa satuan millimeter.

Pada dasarnya, yang tertulis pada LCD adalah sederetan karakter. Oleh karenanya untuk menuliskan angka lebih besar dari pada 9 dilakukan dengan cara menuliskan angka dengan derajat yang paling tinggi, diikuti dengan angka yang derajatnya lebih rendah, sampai akhirnya menuliskan angka yang nilainya satuan. Dengan demikian angka yang lebih besar dari pada 9 harus diuraikan terlebih dahulu menjadi angka satuan, puluhan, ratusan, ribuan, dan seterusnya, sebelum angka tersebut dituliskan ke LCD.

Penulisan angka dalam hal ini sebaiknya dilakukan oleh suatu fungsi tersendiri. Fungsi tersebut harus mempunyai parameter yang berupa angka yang akan dituliskan pada LCD. Angka yang dimasukan

kedalam fungsi akan diuraikan menjadi angka satuan, angka puluhan, angka ratusan, angka ribuan, dan seterusnya sebelum angka-angka tersebut dituliskan pada LCD. Contoh fungsi untuk menuliskan angka pada LCD dapat dilihat pada tabel 3.2. Bentuk Rangkaian LCD dapat dilihat pada gambar 3.4.

**Tabel 3.2**  
**Fungsi Menuliskan Angka Pada LCD**

```
tulis_angka(int angka)
{
  int satuan, puluhan, ratusan, ribuan, sisa;
  if(angka>999)
  {
    ribuan=angka/1000;
    sisa=angka%1000;
    ratusan=sisa/100;
    sisa=sisa%100;
    puluhan=sisa/10;
    satuan=sisa%10;
    lcd_putchar(ribuan+48);
    lcd_putchar(ratusan+48);
    lcd_putchar(puluhan+48);
    lcd_putchar(satuan+48);
  }

  if(angka>99 && angka<=999)
  {
    ratusan=angka/100;
    sisa=angka%100;
    puluhan=sisa/10;
    satuan=sisa%10;
    lcd_putchar(ratusan+48);
    lcd_putchar(puluhan+48);
    lcd_putchar(satuan+48);
  }

  if(angka>9 && angka<=99)
  {
    puluhan=angka/10;
    satuan=angka%10;
    lcd_putchar(puluhan+48);
    lcd_putchar(satuan+48);
  }

  if(angka<=9) lcd_putchar(angka+48);
}
```

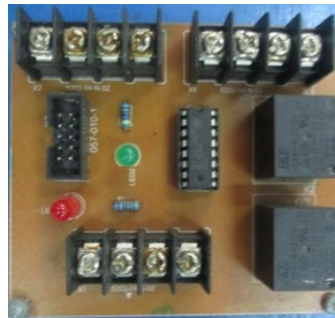


**Gambar 3.4 Rangkaian LCD 16\*2**

### **3.4. Driver Pompa Air**

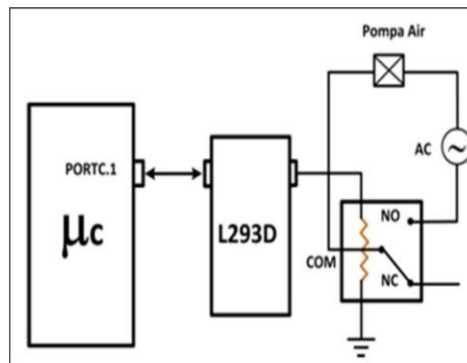
*Driver* pompa air adalah rangkaian yang digunakan untuk menggerakkan pompa air. Driver pompa air akan aktif apabila ada perintah dari mikrokontroler. Rangkaian driver pompa air terdiri dari beberapa komponen elektronika diantaranya IC L293D, *relay*, *Light Emitting Dioda*

(LED), dan resistor. Rangkaian driver pompa dapat dilihat pada gambar 3.6.



**Gambar 3.5 Rangkaian *Driver* Pompa Air**

Arus yang keluar dari mikrokontroler tidak bisa langsung digunakan untuk mengaktifkan *relay*, oleh karena itu perlu komponen yang dapat menguatkan arus sehingga dapat mengaktifkan *relay*. Komponen tersebut adalah IC L293D. Gambar skematik rangkaian *driver* pompa air dapat dilihat pada gambar 3.7.



**Gambar 3.6 Skematik Rangkaian *Driver* Pompa Air**

Prinsip kerja *driver* pompa yaitu ketika input L293D menerima sinyal *high* dari *output* mikrokontroler, sinyal tersebut diteruskan melalui output L293D menuju kaki koil *relay*. Jika koil relay teraliri arus maka kontak relay yang semula terjadi antara COM dan NC akan berubah menjadi antara COM dengan NO. Perubahan dari kontak relay tersebut mengakibatkan pompa air aktif.

## **BAB IV PENGUJIAN DAN ANALISA**

Pada bab ini dibahas tentang pengujian alat penanggulangan banjir di laboratorium otomasi dan analisa hasil pengujian

### **4.1 Pengujian Sensor Ultrasonik Ping Parallax**

Pengujian sensor ultrasonik bertujuan untuk mengetahui perbandingan antara penunjukan LCD terhadap jarak sebenarnya. Pengujian dilakukan dengan cara menempatkan sensor ultrasonik dan penggaris (mistar) sebagai alat ukur jarak atau tinggi terhadap objek yang diletakkan di depan sensor. Pengujian pada sensor ultrasonik dimulai dengan membuat program pada mikrokontroller. *List* program untuk pengujian sensor ultrasonik dapat dilihat pada tabel 4.1.

**Tabel 4.1  
Program Pengujian Sensor Ultrasonik**

```
while (1)
{
  // Place your code here
  counter=0;
  DirSig=1;
  SigOut=1;
  delay_us(5);
  SigOut=0;
  DirSig=0;
  SigOut=1;
  while (SigIn==0) {}
  while (SigIn==1)
  {counter++;}

  distance=(counter*0.034442);

  lcd_clear();
  lcd_gotoxy(2,0);
  lcd_putsf("Jarak");
  lcd_gotoxy(4,1);
  tulis_angka(distance);
  delay_ms(200);
}
```

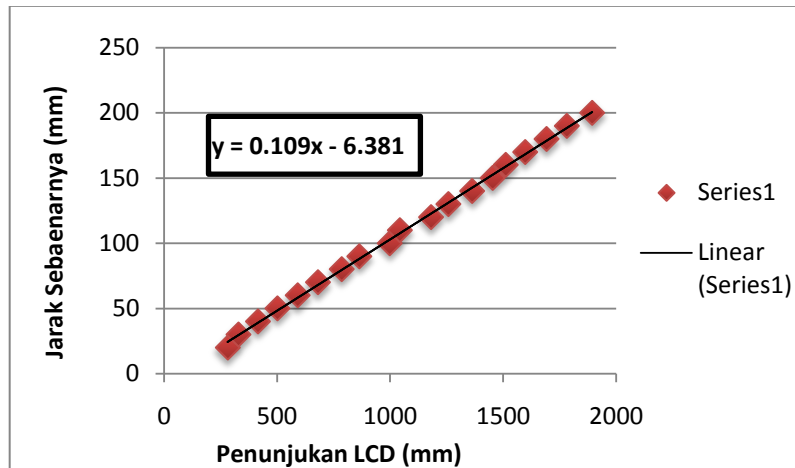
Setelah program tersebut dituliskan secara lengkap program di-*download* ke mikrokontroller. Kemudian objek diletakan didepan sensor

ultrasonik. Objek diletakan diatas penggaris per 10 mm secara berulang – ulang sampai dengan jarak 200 mm. Hasil pengukuran ultrasonik terhadap objek yang diletakan pada penggaris ditampilkan pada LCD. Dari pengujian tersebut dapat diketahui perbandingan jarak yang ditunjukkan pada LCD dengan jarak objek pada penggaris. Tabel hasil pengujian sensor ultrasonik dengan jarak sebenarnya dapat dilihat pada tabel 4.2.

**Tabel 4.2**  
**Hasil Pengujian Sensor Ultrasonik**

<b>NO</b>	<b>JARAK (mm)</b>	<b>LCD</b>
1	20	281
2	30	329
3	40	415
4	50	501
5	60	591
6	70	681
7	80	786
8	90	863
9	100	998
10	110	1043
11	120	1181
12	130	1258
13	140	1363
14	150	1453
15	160	1511
16	170	1597
17	180	1692
18	190	1782
19	200	1894

Setelah diketahui jarak yang ditunjukkan pada LCD, data kemudian diolah pada kurva jarak penunjukan pada LCD vs jarak sebenarnya. Dari kurva tersebut didapat persamaan penunjukan pada LCD vs jarak sebenarnya dengan satuan (mm) yang nantinya akan digunakan sebagai konversi dari jarak sensor ultrasonik terhadap objek. Kurva penunjukan pada LCD vs jarak sebenarnya dapat dilihat pada gambar 4.1.



**Gambar 4.1**  
**Kurva Penunjukan LCD vs Jarak Sebenarnya**

Setelah diketahui persamaan penunjukan pada LCD vs jarak sebenarnya, selanjutnya persamaan tersebut dimasukkan kedalam program. *List* program dapat dilihat pada tabel 4.3.

**Tabel 4.3**  
**List Program Dengan Menggunakan Persamaan**

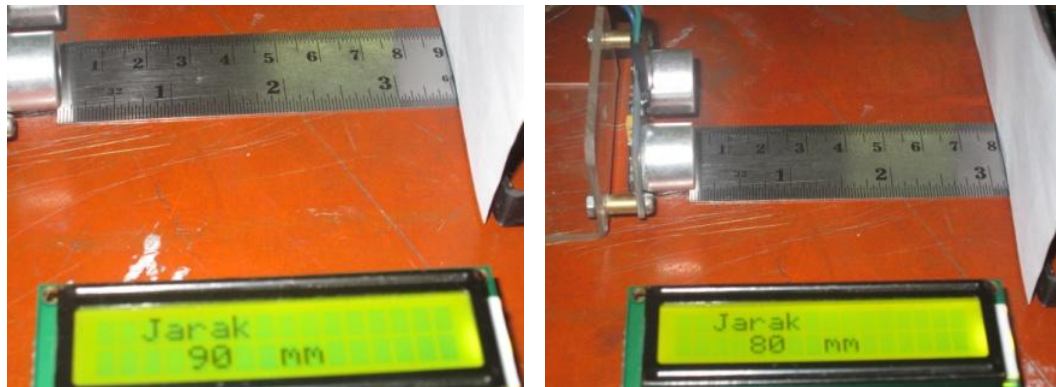
```
while (1)
{
// Place your code here
counter=0;
DirSig=1;
SigOut=1;
delay_us(5);
SigOut=0;
DirSig=0;
SigOut=1;
while (SigIn==0){ }
while (SigIn==1)counter++;

distance=(counter*0.109-6.381);

if(distance>=385)PORTC.1=0;
if(distance<=360)PORTC.1=1;

lcd_clear();
lcd_gotoxy(2,0);
lcd_putsf("Jarak");
lcd_gotoxy(4,1);
tulis_angka(distance);
lcd_gotoxy(8,1);
lcd_putsf("mm");
delay_ms(200);
}
```

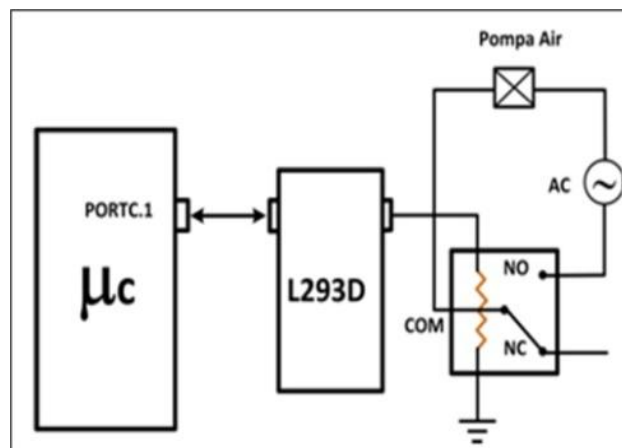
Setelah memasukan persamaan pununjukan LCD vs jarak sebenarnya kedalam program dengan benar, program dapat di-download ke mikrokontroller. Hasil dari eksekusi program tersebut didapatkan keakuratan antara penunjukan LCD terhadap jarak sebenarnya. Gambar keakuratan penunjukan LCD terhadap jarak sebenarnya dapat dilihat pada gambar 4.2.



**Gambar 4.2**  
**Keakuratan Penunjukan LCD Terhadap Jarak Sebenarnya**

#### **4.2 Pengujian *Driver* Pompa Air**

Pengujian *driver* pompa air bertujuan untuk mengetahui apakah rangkaian *driver* dapat berfungsi atau tidak untuk menghidupkan pompa air sesuai ketinggian permukaan air terhadap sensor di penampungan. Gambar skematik rangkaian *driver* pompa air dapat dilihat pada gambar 4.3.



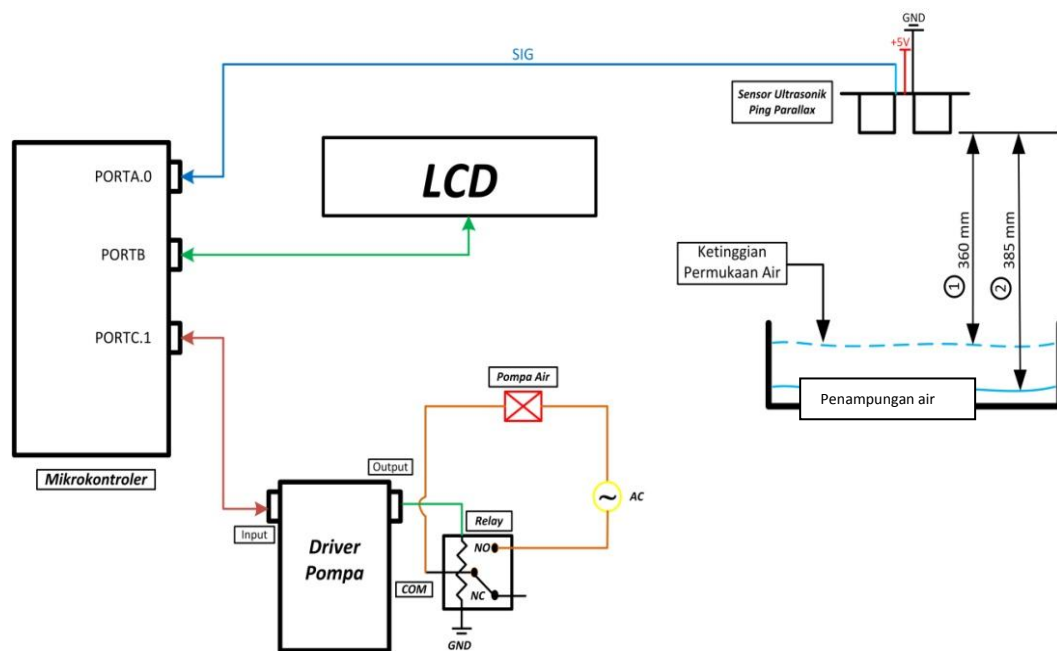
**Gambar 4.3 Skematik *Driver* Pompa Air**



Pengujian *driver* pompa air dilakukan dengan cara menghubungkan *driver* pompa air dengan mikrokontroller pada PORTC.1. PORTC.1 pada mikrokontroller diberi nilai *high* atau *low*. Jika PORTC.1 pada mikrokontroller diberi nilai *high* maka *driver* dan pompa air menjadi aktif. Jika PORTC.1 pada mikrokontroller diberi nilai *low* maka *driver* pompa tidak aktif dan hasil dari penkondisian tersebut mengakibatkan *driver* dan pompa air menjadi tidak aktif.

### 4.3 Pengujian Alat Keseluruhan

Setelah semua rangkaian dipastikan dalam kondisi baik, dilakukan pengujian dari alat keseluruhan dan diaplikasikan pada penampungan air. Gambar skematik rangkaian keseluruhan dapat dilihat pada gambar 4.4.



**Gambar 4.4**

### Rangkaian Keseluruhan Alat penanggulangan Banjir

Pengujian dilakukan dengan cara mengalirkan air kedalam penampungan. Sensor ultrasonik yang diletakan diatas penampungan air memancarkan gelombang ultrasonik melalui *transmitter* dan terpantul kembali menuju *receiver* ketika menyentuh permukaan air. Mikrokontroller mendata bilangan dari sensor ultrasonik sesuai dengan penginputan untuk

menghidupkan pompa air. Data perubahan ketinggian permukaan air di tampilkan pada LCD. Pada saat sensor ultrasonik terhadap permukaan air berjarak 360 mm mengakibatkan *driver* pompa aktif dan pompa air menjadi aktif. Gambar penunjukan angka 360 mm pada LCD dapat dilihat pada gambar 4.5.



**Gambar 4.5**  
**Penunjukan Angka 360 mm Pada LCD**

Pada saat pompa air aktif mengakibatkan air di penampungan surut. Ketika jarak antara sensor ultrasonik terhadap permukaan air dipenampungan sekitar 385 mm *driver* pompa tidak aktif dan pompa air tidak aktif. Gambar penunjukan angka 385 mm pada LCD dapat dilihat pada gambar 4.6.



**Gambar 4.6**  
**Penunjukan Angka 385 mm Pada LCD**

#### **4.4 Analisa Hasil Pengujian**

Dari hasil pengujian jarak pada sensor ultrasonik yang telah dilakukan didapat kurva penunjukan pada LCD dengan jarak sebenarnya. Dari kurva tersebut didapat persamaan yang selanjutnya dijadikan sebagai konversi antara pembacaan dari sensor ultrasonik terhadap penunjukan pada LCD.

Hasil dari pengujian alat keseluruhan juga dapat dianalisa beberapa hal yaitu :

1. *Driver* pompa dan pompa air aktif pada jarak antara sensor ultrasonik terhadap permukaan air berjarak 360 mm.
2. *Driver* pompa dan pompa tidak aktif pada jarak antara sensor ultrasonik terhadap permukaan air berjarak 385 mm.

## **BAB V KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan dan saran setelah melakukan pengujian dan analisa hasil pengujian.

### **5.1. Kesimpulan**

Setelah dilakukan pengujian dan analisa, dapat disimpulkan bahwa alat penanggulangan banjir di laboratorium otomasi yang dibuat sudah berfungsi dengan baik, dimana satuan hasil pengukuran adalah mm (mili meter).

### **5.2. Saran**

Dalam pembuatan alat penanggulangan banjir di laboratorium otomasi tidak terlepas dari beberapa kekurangan. Saran ini bertujuan agar kinerja alat tersebut dapat ditingkatkan dan dikembangkan lagi, dengan cara menambahkan komponen seperti komponen *buzzer* dan lampu tanda bahaya.

**DAFTAR PUSTAKA**

1. Hartono, Rachmad. 2013 : Diktat Belajar Mikrokontroler, Bandung, Universitas Pasundan.
2. Hartono, Rachmad. 2013 : Komponen Elektronika, Bandung, Universitas Pasundan.
3. Pengujian Sensor Ultrasonik PING Untuk Pengukuran Level Ketinggian dan Volume Air, data diperoleh melalui situs internet : [www.allbookez.com](http://www.allbookez.com). 20 Desember 2014.
4. PING))) Ultrasonic Distance Sensor, data diperoleh melalui situs internet : [www.cypress.com](http://www.cypress.com). 14 Oktober 2014.
5. Pratama, Agung, M. 2014 : Perancangan Dan Pembuatan *Wrist Assembly* Pada Robot Lengan, Bandung. Universitas Pasundan, Bab IV : 1-7.

## **L1. Program Pengujian alat Penanggulangan Banjir Di Laboratorium Otomasi**

This program was created by the  
CodeWizardAVR V3.04 Evaluation  
Automatic Program Generator  
© Copyright 1998-2013 Pavel Haiduc, HP InfoTech s.r.l.  
<http://www.hpinfotech.com>

Date : 12/4/2013  
Author : Dodi Hidayat  
Company : Peotro Labs.  
Comments:

Chip type : ATmega8535L  
Program type : Application  
AVR Core Clock frequency : 16.000000 MHz  
Memory model : Small  
External RAM size : 0  
Data Stack size : 128

```
#include <mega8535.h>
#include <delay.h>
#include <stdio.h>
```

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

```
// Declare your global variables here
#define SigOut PORTA.0
#define SigIn PINA.0
#define DirSig DDRA.0
```

```
unsigned int counter;
float distance;
//unsigned char Baris1[16];
//unsigned char Baris2[16];
```

```
tulis_angka(int angka)
{
    int satuan, puluhan, ratusan, ribuan, sisa;
    if(angka>999)
    {
        ribuan=angka/1000;
        sisa=angka%1000;
        ratusan=sisa/100;
        sisa=sisa%100;
```

## **Laporan Tugas Akhir**

---

```
puluhan=sisa/10;
satuan=sisa%10;

lcd_putchar(ribuan+48);
lcd_putchar(ratusan+48);
lcd_putchar(puluhan+48);
lcd_putchar(satuan+48);
}

if(angka>99 && angka<=999)
{
    ratusan=angka/100;
    sisa=angka%100;
    puluhan=sisa/10;
    satuan=sisa%10;

    lcd_putchar(ratusan+48);
    lcd_putchar(puluhan+48);
    lcd_putchar(satuan+48);
}

if(angka>9 && angka<=99)
{
    puluhan=angka/10;
    satuan=angka%10;

    lcd_putchar(puluhan+48);
    lcd_putchar(satuan+48);
}

if(angka<=9) lcd_putchar(angka+48);

}

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

    // Input/Output Ports initialization
    // Port A initialization
    // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
    DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) |
    (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
    // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
    PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) |
    (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

    // Port B initialization
```

## **Laporan Tugas Akhir**

---

```
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) |
(0<<DDB2) | (0<<DDB1) | (0<<DDB0);
```

```
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) |
(0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);
```

```
// Port C initialization
```

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

```
DDRC=(1<<DDC7) | (1<<DDC6) | (1<<DDC5) | (1<<DDC4) | (1<<DDC3) |
(1<<DDC2) | (1<<DDC1) | (1<<DDC0);
```

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

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

```
// Port D initialization
```

```
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) |
(0<<DDD2) | (0<<DDD1) | (0<<DDD0);
```

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

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

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

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

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

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

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

```
TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) |
(0<<CS02) | (0<<CS01) | (0<<CS00);
```

```
TCNT0=0x00;
```

```
OCR0=0x00;
```

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

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

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

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

```
// OC1A output: Disconnected
```

```
// OC1B output: Disconnected
```

```
// Noise Canceler: Off
```

```
// Input Capture on Falling Edge
```

```
// Timer1 Overflow Interrupt: Off
```

```
// Input Capture Interrupt: Off
```

```
// Compare A Match Interrupt: Off
```

```
// Compare B Match Interrupt: Off
```

```
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) |
(0<<WGM11) | (0<<WGM10);
```



## **Laporan Tugas Akhir**

---

```
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) |  
(0<<CS12) | (0<<CS11) | (0<<CS10);  
TCNT1H=0x00;  
TCNT1L=0x00;  
ICR1H=0x00;  
ICR1L=0x00;  
OCR1AH=0x00;  
OCR1AL=0x00;  
OCR1BH=0x00;  
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization  
// Clock source: System Clock  
// Clock value: Timer2 Stopped  
// Mode: Normal top=0xFF  
// OC2 output: Disconnected  
ASSR=0<<AS2;  
TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) |  
(0<<CS22) | (0<<CS21) | (0<<CS20);  
TCNT2=0x00;  
OCR2=0x00;
```

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

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

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

```
// SPI initialization  
// SPI disabled  
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) |  
(0<<CPHA) | (0<<SPR1) | (0<<SPR0);
```

## ***Laporan Tugas Akhir***

---

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

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTB Bit 0
// RD - PORTB Bit 1
// EN - PORTB Bit 2
// D4 - PORTB Bit 4
// D5 - PORTB Bit 5
// D6 - PORTB Bit 6
// D7 - PORTB Bit 7
// Characters/line: 16
lcd_init(16);

while (1)
{
    // Place your code here
    counter=0; //initial value
    DirSig=1; //set as output
    SigOut=1;
    delay_us(5);
    SigOut=0;
    DirSig=0; //set as input
    SigOut=1; //Pullup activated
    while (SigIn==0){}
    while (SigIn==1)counter++;

    distance=(counter*0.109-6.381);

    if(distance>=385)PORTC.1=0;
    if(distance<=360)PORTC.1=1;

    lcd_clear();
    lcd_gotoxy(2,0);
    lcd_putsf("Jarak");
    lcd_gotoxy(4,1);
    tulis_angka(distance);
    lcd_gotoxy(8,1);
    lcd_putsf("mm");
    delay_ms(200);
}
}
```

## **L2. Percobaan Dengan Pelampung Air (RADAR ST-70AB)**



Jarak ketinggian air dalam wajan atau ember dengan ketinggian air 150 mm, mengakibatkan lampu AC menyala. Penggunaan pelampung air tidak efektif apabila diaplikasikan di penampungan air, karena jarak yang diperlukan untuk menghidupkan lampu atau pompa air yaitu 90 mm.

## **L3. Rangkaian Kontrol Alat Penanggulangan Banjir Di Laboratorium Otomasi**



**LAMPIRAN**

