### BAB V

# IMPLEMENTASI DAN PENGUJIAN SISTEM

### 5.1 HASIL IMPLEMENTASI

Pada tahap ini penulis mengimplementasikan hasil rancangan yang telah dibuat. Adapun hasil implementasi dari penelitian ini dapat dilihat pada gambar 5.1.





Gambar diatas merupakan bentuk fisik dari perancangan perangkat *monitoring* kualitas air pada budidaya kolam ikan berbasis sms getway yang telah dirancang penulis. Terdapat beberapa komponen yang ada pada alat ini yaitu, sistem minimum sebagai pengendali dari semua komponen yang ada, sensor sensor pH meter digital yang berfungsi untuk alat ukur yang digunakan untuk

mengukur pH air pada kolam, sms getway berfungsi untuk memberi peringatan kepada user jika terjadi keasaman (pH) air berlebih serta satu buah LCD 16 x 2 sebagai pemberi informasi tentang nilai keasaman (pH) air.

Pada gambar 5.1, dan 5,2 adalah cara kerja sistem alat yaitu apabila sensor pH membaca kadar air yang mempunyai keasaman berlebih pada kolam ikan, maka sensor akan mengirim nilai ke dalam mikrokontroller, dan dilanjutakan lagi dari mikrokontroller ke user mealui sms.



### Gambar 5.2 Kadar Air Keasaman Berlebih

Pada gambar 5.2 apabila nilai kadar air diatas 8.6 menandakan bahwa kebasahan air meningkat, maka mikrokontroller akan memberi peringatan atau pesan kepada user jika terjadi kebasahan (pH) air berlebih.

### 5.2 PENGUJIAN PERANGKAT LUNAK

Hal pertama yang dilakukan dalam pengujian perangkat lunak adalah menentukan aplikasi yang akan digunakan untuk mengisi program pada mikrokontroler Atmega16.

#### A. CodeVision AVR

Untuk mikrokontroler ATmega16 sinkron dengan banyak bahasa pemrograman seperti *Assembler, CodeVision AVR*, bahasa C, *BASCOM AVR* dan lainnya. Dalam pembuatan alat ini penulis menggunakan CodeVision AVR, sebab CodeVision AVR sangat kompatibel dengan downloader yang penulis gunakan.

Untuk pengujian yaitu pembuatan listing program baru, tekan file kemudian pilih *Project* 



Gambar 5.3 Menu Membuat Program Baru

Maka selanjutnya akan muncul kembali jendela konfirmasi project, pilih

Yes.



Gambar 5.4 Konfirmasi Project

Kemudian akan muncul kembali jendela *Wizard AVR*, pada bagian tentukan *chip* yang akan digunakan. Karena *chip* yang akan digunakan maka ceklis pilihan yang pertama AT90, Attiny, Atmega lalu klik ok.

👍 CodeWizardAVR
Target AVR Chip Type
AT90, ATtiny, ATmega
© ×MEGA
✓ <u>D</u> K X <u>C</u> ancel

Gambar 5.5 CodeWizardAVR

Berikutnya masuk ke dalam jendela *CodeWizardAVR*. Di jendela ini terdiri dari beberapa bagian tab pilihan yaitu terdiri dari :

1. Tab Chip

Yaitu *Tab* yang menentukan pilihan *Chip* yang digunakan Atmega16,

Clock yang digunakan 12.000.000 MHz.



Gambar 5.6 Tab Chip

2.

Tab Ports

Port A : Pada port ini yaitu A0 digunakan sebagai *input* dari sensor *pH* meter digital. Pada *ports setting data direction* tetap In dan Pullup/Output Value T.

CodeWizardAVR - untitled.cwp X					
<u>F</u> ile <u>H</u> elp	0				
12C	11	Wire		2 Wire	e (12C)
LCD	Bit-Bang	ged	Pro	ject Info	ormation
USART	Analog C	omparat	or	ADC	SPI
Chip	Ports	Exter	mal	IRQ	Timers
Port A	Port B	Port C	Po	ort D	
D	ata Direct	ion P	ullu	p/Outp	ut Value
	Bit 0 _1	n	Т	Bit 0	
	Bit 1 _I	n	Т	Bit 1	
	Bit 2 _1	n	Τ	Bit 2	
	Bit 3 _I	n	T	Bit 3	
	Bit 4	n	Т	Bit 4	
	Bit 5_I	n	T	Bit 5	
	Bit 6 _I	n	Т	Bit 6	
	Bit 7 _1	n	Т	Bit 7	

Gambar 5.7 Input Sensor Ph Meter Digital Pada Port A

Port B : pada port B digunakan untuk untuk usbasp, downloader ini memiliki 5 kabel yaitu miso, mosi, sck, dan reset. miso dihubungkan ke pin miso mikrokontroller, mosi ke pin mosi mikrokontroller, sck ke pin sck, dan rst ke pin reset.

Izu     I Wire     2 Wire (IzU)       Dit-Banged     Project Information       Chip     Ports     External IRQ       Data Direction     Pullup/Output Value       Bit 0     In     T       Bit 1     In     T       Bit 2     In     T       Bit 3     In     T       Bit 4     In     T       Bit 5     In     T       Bit 6     In     T	100	Analog	Compar	ator	ADC	300
LU     Bit-Eanged     Project Information       Ihip     Ports     External IRQ     Timer       Port A     Port B     Port C     Port D       Data Direction     Pullup/Output Value       Bit 0     In     T     Bit 1       Bit 2     In     T     Bit 2       Bit 3     In     T     Bit 3       Bit 4     In     T     Bit 4       Bit 5     In     T     Bit 5       Bit 6     In     T     Bit 6       Bit 7     In     T     Bit 7	120	D.2 D	l Wire	2 Wire (I		re (I2C)
Port A     Port B     Port C     Port D       Data Direction     Pullup/Output Value       Bit 0     In     T     Bit 0       Bit 1     In     T     Bit 1       Bit 2     In     T     Bit 2       Bit 3     In     T     Bit 3       Bit 4     In     T     Bit 4       Bit 5     In     T     Bit 5       Bit 6     In     T     Bit 6       Bit 7     In     T     Bit 7		Bit-Ba	nged	Pro	ject in	formation
Port A         Port B         Port C         Port D           Data Direction         Pullup/Output Value           Bit 0         In         T         Bit 0           Bit 0         In         T         Bit 1           Bit 1         In         T         Bit 1           Bit 2         In         T         Bit 3           Bit 4         In         T         Bit 3           Bit 4         In         T         Bit 4           Bit 5         In         T         Bit 5           Bit 6         In         T         Bit 7           Bit 7         In         T         Bit 7	Lhip	Fulls	Ext	ernal	IRQ	l imer:
Data Direction         Pullup/Output Value           Bit 0         _In         _I         Bit 0           Bit 1         _I         _I         Bit 1           Bit 2         _In         _I         Bit 2           Bit 3         In         _I         Bit 3           Bit 4         _In         _I         Bit 4           Bit 5         _In         _I         Bit 5           Bit 6         _In         _I         Bit 6           Bit 7         _In         _I         Bit 7	Port A	Port B	Port C	Po	ort D	
Bit 0 _ In _ T Bit 0 Bit 1 _ In _ T Bit 1 Bit 2 _ In _ T Bit 3 Bit 4 _ In _ T Bit 4 Bit 5 _ In _ T Bit 4 Bit 5 _ In _ T Bit 5 Bit 6 _ In _ T Bit 7	(	_ Data Dire	ction	Pullu	p/Outp	put Value
Bit 1 _ In _ T Bit 1 Bit 2 _ In _ T Bit 3 Bit 3 _ In _ T Bit 3 Bit 4 _ In _ T Bit 4 Bit 5 _ In _ T Bit 5 Bit 6 _ In _ T Bit 6 Bit 7 _ In _ T Bit 7		Bit 0	In	Τ	Bit 0	
Bit 2 _ in _ T Bit 2 Bit 3 _ in _ T Bit 3 Bit 4 _ in _ T Bit 4 Bit 5 _ in _ T Bit 5 Bit 6 _ in _ T Bit 6 Bit 7 _ in _ T Bit 7		Bit 1	In	Τ	Bit 1	
Bit 3 _ in _ T Bit 3 Bit 4 _ in _ T Bit 4 Bit 5 _ in _ T Bit 5 Bit 6 _ In _ T Bit 5 Bit 7 _ in _ T Bit 7		Bit 2	In	Τ	Bit 2	
Bit 4inT Bit 4 Bit 5inT Bit 5 Bit 6inT Bit 6 Bit 7inT Bit 7		Bit 3	In	Т	Bit 3	
Bit 5 <u>in</u> <u>T</u> Bit 5 Bit 6 <u>in</u> <u>T</u> Bit 6 Bit 7 <u>in</u> <u>T</u> Bit 7		Bit 4	In	Τ	Bit 4	
Bit 6 <u>In</u> <u>T</u> Bit 6 Bit 7 <u>In</u> <u>T</u> Bit 7		Bit 5	In	Τ	Bit 5	
Bit 7 _ In _ T Bit 7		Bit 6	In	Т	Bit 6	
		Bit 7	In	Т	Bit 7	

Gambar 5.8 Menentukan Port B

Port C : Pada port ini C0 – C7 digunakan sebagai *output* dari LCD 2x16.

Pada ports setting data direction In diubah menjadi Out.

CodeWizardAVR - untitled.cwp ×				
USART 120	Analog Compar	ator	ADC 2 Wire	SPI
Chip	Ports Ex	ernal IRQ Timer		Timers
LCD	Bit-Banged	Pro	ject Info	rmation
	LCD Port: PC Chars./Line: 18 0RT Bit 0 - RS 0RT Bit 1 - RD 0RT Bit 2 - EN 0RT Bit 3 - Fire 0RT Bit 4 - DB 0RT Bit 5 - DB 0RT Bit 5 - DB 0RT Bit 7 - DB		Pin 4) Pin 5) Pin 6) Pin 12 Pin 12 Pin 13 Pin 14	) () )

Gambar 5.9 Menentukan Port C

#### 3. Tab USART

Pada tab USART digunakan untuk mengaktifkan *transmitter*, *receiver*, dan *baud rate*. Ceklis *receiver* dan *transmitter*, lalu pilih *baudrate* menjadi 9600.

CodeWizardAVR - untitled.cwp × File Help				
12C	1 Wire	2 Wire (I2C)		
Chip	Ports Exte	rnal IRQ Timers		
LCD	Bit-Banged	Project Information		
USART	Analog Compara	tor ADC SPI		
	-			
V	] <u>T</u> ransmitter [	Tx Interrupt		
Baud Rate: 9600 V				
Co	ommunication Para	ameters:		
8	Data, 1 Stop, No	Parity 🗸 🗸		
Mod	e: Asynchronous	$\sim$		

Gambar 5.10 Tab USART

Setelah menentukan tab *chip, port,* dan USART pilih menu pilihan program generate, save and exit. Muncul jendela Save C Compiler Source File yaitu jendela untuk menyimpan file *project* yang akan kita simpan. Pada CodeVision AVR menyimpan *file* terdiri dari tiga kali pemyimpanan. File pertama dengan ekstensi .C, *Prj, .Cwp*.



Gambar 5.11 Jendela Menyimpan File Project

Kemudian setelah menyimpan file project akan muncul jendela koding

program yang berupa koding *default*. Disini mulai pengetikkan *listing* program.

👯 Code	visionAVR - C:\Users\IDEL_DELFRI\Desktop\andi\2.prj						
Eile E	Edit Search View Project Iools Settings Help						
-	Ů▓▾◱◙◳淋淋淋淋™;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;						
R 🔋	: % % O ✔ O O O O A A D P 6 6 D 2 . 4 * # # 0 = 0 . 13 & % # \$ 0 . 0						
中型	i 🗄 👖 🗄 🕼 🖤 🗄 🗖 🧶 🥅 🔲 🚍 🖕						
<b>M</b> H	a 🟥 🥶 🗛 🦽 🖕						
HE C.V.	Jaers/IDEL_DELFRI/Desktop/andh2.c						
8 /	Notes / 2.c 📓						
2 1							
2	This program was created by the						
3	CodeWizzrdAVR V3.12 Advanced						
- 4	Automatic Program Generator						
5	© Copyright 1998-2014 Favel Haiduc, HF InfoTech s.r.l.						
6	http://www.hpinfotech.com						
7							
8	Froject :						
9	Version :						
10	Date : 1/13/2017						
11	Author :						
12	Company :						
13	Comments:						
14							
15							
16	Chip type : ATmegal6						
17	Program type : Application						
18	18 AVR Core Clock frequency: 12.000000 MHz						
19	9 Memory model : Small						
20	External RAM size : 0						
21	Data Stack size : 256						
22							
23							
24	\$include <mega16.h></mega16.h>						

Gambar 5.12 *Listing* Program

Setelah *listing* program dibuat maka tahap berikutnya adalah meng*compile* program yang dibuat atau menguji kebeneran koding - koding program yang kita buat dengan cara menekan CTRL + F9 dan apabila program yang kita buat salah maka akan muncul jendela informasi bahwa terdapat *error*, terdapat petunjuk dimana terjadi kesalahan tersebut, dan apabila program yang kita buat benar maka akan muncul informasi *No error*, maka program langsung meng*compile* program tersebut seperti tampak pada gambar 5.13 dan 5.14 berikut :

0	Inform	nation	x
Γ	Compiler		
	Build conf Chip: ATm Clock freq Program ty Memory m Optimize f (s)printf fe. (s)scanf fe (s)scanf fe (s)scanf fe (s)scanf fe Promote 'c (s)scanf fe (s)scanf fe (s	figuration: Debug mega16 quency: 12.000000 MHz type: Application nodet: Small for: Size astures: int, width eatures: int, width char' to 'int: Yes maigned: Yes mist' stored in FLASH: Yes ms: Yes d function parameters passing: Yes register allocation: Yes jister allocation: Yes	
	Build: 1 1466 line(: 1 error(s) No warnin	(s) compiled ngs	
E	Bit variabl	les size: 0 byte(s)	
	Data Stac Data Stac Estimated	ck area: 0x60 to 0x15F ck size: 256 byte(s) I Data Stack usage: 0 byte(s)	
F	RAM Glob	bal variables size: 0 byte(s)	
ł	Hardware Hardware	e Stack area: 0x160 to 0x45F e Stack size: 768 byte(s)	
ł	Heap size	e: 0 byte(s)	
E	EEPROM	l usage: 0 byte(s), 0.0% of EEPROM	
	🔽 Show	v Information after Build	
			$\langle $

Gambar 5.13 Tampilan Program Ketika Terjadi Error

1 Inform	mation	X
Compiler		
Build conl Chip: ATn Clock fred Program ( Memory m Optimize f (s)scanf fe (s)scanf fe (s)scanf fe Promote ' 'char' is uu global 'co 8 bit enun Enhanced Automatic Smart reg	higuration: Debug mega16 type: Application modet: Small for: Size eatures: int, width features: int, width features: int, width chart to Imit: Yes unsigned: Yes onst stored in FLASH: Yes ms: Yes d function parameters passing: Yes ic register allocation: Yes gister allocation: Yes	
Build: 1 9298 line( No errors No warnir	e(s) compiled s ings	
Bit variabl	bles size: 0 byte(s)	
Data Stac Data Stac Estimated	ack area: 0x60 to 0x15F ack size: 256 byte(s) d Data Stack usage: 0 byte(s)	
RAM Glot	obal variables size: 0 byte(s)	
Hardware Hardware	e Stack area: 0x160 to 0x45F e Stack size: 768 byte(s)	
Heap size	re: 0 byte(s)	
EEPROM	প usage: 0 byte(s), 0.0% of ΕΕΡRΟΜ	
Show	w Information after Build	<u>0</u> K

Gambar 5.14 Tampilan Program Tidak Terdapat Error

Selain pengujian *software*, penulis juga melakukan pengujian dari aplikasi yang dibuat yaitu *visual basic*. Pengujian yang dilakukan adalah mengecek komunikasi serial apakah alat sudah terhubung atau belum ke komputer atau laptop, seperti tampak pada tabel 5.1.

## 5.3 PENGUJIAN ALAT

### 5.3.1 Pengujian Tegangan Sumber

Tahap pertama yang dilakukan adalah pengecekan catu daya. Catu daya yang digunakan memiliki keluaran sebesar 12 volt. Pengujian dilakukan dengan cara menggunakan multimeter. Hubungkan katup positif dari multimeter ke keluaran 12 volt dan hubungkan katup negatif multimeter ke ground pada catu daya.

Hasil pengujian tegangan catu daya 12 volt dapat dilihat pada tabel 5.1

Tabel 5.1 Pengujian Tegangan Catu Daya 12 Volt

Sumber Arus	Tegangan Input	Tegangan Output
Catu Daya	12 Volt	11.8 Volt

#### 5.3.2 Pengujian Tegangan ATmega16

Rangkaian ini merupakan otak dari seluruh rangkaian. Semua rangkaian yang ada dikendalikan *input output*-nya oleh rangkaian mikrokontroler ini. Proses pengujian rangkaian ini adalah dengan menghubungkan pin VCC (+) dan GND (-) pada multimeter. Adapun hasil dari pengujian tegangan ATmega16 ini dapat dilihat pada tabel 5.2.

Tabel 5.2 Pengujian Tegangan ATmega16

Sumber Arus	Tegangan Input	Tegangan Output
ATmega	5 Volt	4.8 Volt

#### 5.3.3 Pengujian Downloader

Downloader berfungsi untuk mengisi program ke alat. Pada pengujian downloader hal yang dilakukan adalah menghubungkan downloader ke komputer atau laptop. Downloader memiliki 6 pin yang akan dihubungkan ke ATmega16 terdiri dari MOSI, MISO, SCK, RST, VCC dan GND. Untuk mengecek apakah downloader sudah terhubung dengan baik, pada aplikasi CodeVision AVR pilih menu *tools – chip programmer*. Akan tampil tab CodeVision AVR *chip programmer*. Pilih menu *read – chip signature*, apabila downloader berfungsi dengan baik maka akan tampil tab *information*, bila downloader tidak terhubung maka akan tampil tab *error* dapat dilihat pada gambar 5.15, 5.16 dan 5.17.

le Edit Program Re	ad Compare	Help				
Chip: ATmega16	•	📖 Progr	am <u>A</u> ll	•	Re <u>s</u> et C	hip
FLASH Start: 0 h End Checksum: 0x1FBC	405 h	EEPRON Start: 0 Checksu	4 h m: 0xFE00	End:	1FF <sup>9</sup> rogram	h
Chip Programming Optic FLASH Lock Bits No Protection Programming dise Programming and Boot Lock Bit 0 BO1=1 B02=1 B01=0 B02=1 B01=0 B02=0 B01=1 B02=0	abled I Verification disal Boot Lock Bi B11=1 B1 B11=0 B1 B11=0 B1 B11=1 B1 B11=1 B1	oled [ 1 2=1 [ 2=0 [	Program CKSEL CKSEL CKSEL SUTI= BODEN BODEN BOOTS BOOTS BOOTS EESAV CKOPT JTAGE OCDEN	Fuse B 0=0 1=0 2=0 0 4=0 VEL=0 iZ0=0 iZ1=0 iZ1=0 N=0 i=0	) 1	
📝 Check Signature	🗖 Check Erasur	e 🔲 Pre:	serve EEF	ROM	📝 Veri	ify

Gambar 5.15 Tab CodeVision AVR Chip Programmer



Gambar 5.16 Tampilan Downloader Sudah Terhubung



Gambar 5.17 Tampilan Downloader Tidak Terhubung

# 5.3.4 Pengujian Sensor *pH*

Pengujian terhadap sensor pH dilakukan dengan cara menghubungkan ke pengkabelan dari pH ke mikrokontroller atmega 16 berupa po, ground dan vcc sedangkan hasil keluarannya ditampilkan pada LCD 16 x 2.





Pengujian sensor pH ini dilakukan untuk mengukur kandungan air dalam range 0-14 dan respon pembacaan sensor saat bekerja. Berikut hasil pengujian sensor pH dapat dilihat pada table 5.3

Percobaan ke	Kandungan <i>pH</i>
1	7.54
2	7.55
3	7.60
4	7.40

Tabel 5.3 Pengujian Sensor *pH* 

Untuk mendapatkan keluaran dari sensor pH, dapat dilakukan perhitungan dengan persamaan sebagai berikut:

### pH = (float)adc\*5.0/1024/6

Pengujian pertama sensor pH dilakukan dengan cara pengkalibrasian suatu nilai sensor pH yang digunakan dengan nilai pH di dalam larutan standar buffer yang sudah diukur dengan alat lain. Pengujian ini juga bertujuan untuk mengetahui perubahan pembacaan nilai pH terhadap waktu. Tabel 4.1 ditunjukkan data hasil pengujian pH setiap 10 detik sampai 70 detik dengan perbandingan nilai pH buffer standar 6 dan 7.

Tabel 5.4 Kalibrasi larutan buffer standard dengan nilai pH7

No	Pengukuran dengan Sensor <i>pH</i> yang digunakan	Waktu (Detik)
1	7.35	10
2	7.50	20
3	7.55	30
4	7.60	40
5	7.30	50
6	7.40	60
7	7.50	70

#### 5.3.5 Pengujian Module Sms (SIM800L)

Pengujian Module SIM800L ini berdasarkan data dari sensor pH yang telah diproses oleh mikrokontroller. Data yang dikirim berupa sms yang berdasarkan hasil dari nilai sensor pH dan dikirim ke nomor *handphone* yang telah dicantum didalam program. Tabel hasil pengujian module sim800L dapat dilihat dalam tabel 5.5 :

No	Nilai Senor pH	SMS
1	8.61	Kirim sms
2	6.69	Kirim sms
3	6.7 - 8.6	Tidak Kirim Sms

 Tabel 5.5 Hasil Pengujian Module SIM800L

Dari hasil pengujian di tabel 5.5, dapat disimpulkan bahwa modul sms SIM800L berfungsi dengan baik.

### 5.4 ANALISIS SISTEM SECARA KESELURUHAN

Untuk mendeteksi apabila terjadi kesalahan setelah uji coba, maka perlu dilakukan analisa rangkaian secara keseluruhan. Dari seluruh proses yang telah dilakukan, baik pengujian perangkat keras maupun perangkat lunak, dapat dikatakan bahwa alat ini dapat berfungsi sebagaimana yang penulis inginkan.

Proses pembacaan sensor pH tidak terjadi kesalahan pembacaan data, Pengujian ini dilakukan untuk menunjukan bahwa alat *monitoring* kualitas air ini dapat bekerja sesuai dengan tujuan dari pembuatan. Pengujian ini dilakukan dengan cara sebagai berikut :

- 1. Letakan sensor pH pada air yang akan dipantau.
- Kemudian LCD 16 x 2 akan menampilkan nilai yang didapat dari sensor *pH* melalui mikrokontroller.
- 3. Jika nilai Sensor *pH* diatas 8.610 dan kurang dari 6.69 maka mikrontroller akan mengirim sms kepada user bahwa terjadi keasaman pada kolam ikan.