

BAB V

PENUTUP

A. Kesimpulan

Setelah dilakukan pengujian dan analisa program, maka dapat diperoleh kesimpulan :

1. RTC (*Real Time Clock*) ditambahkan sebagai pengatur waktu otomatis yang dapat terjadwal.
2. Mikrokontroler AT Mega 16 memberikan logika 0 atau 1 ke bagian relay, kemudian relay meneruskan sinyal tersebut ke solenoid valve sedangkan logika 1 digunakan untuk menutup solenoid valve.
3. Pengaturan waktu penyiraman dan penggantian air pada media tanam hidroponik menggunakan RTC (*Real Time Clock*) yaitu memanfaatkan program pengkondisian ketika waktu menunjukkan jam, menit dan detik tertentu maka mikrokontroler akan mengaktifkan solenoid valve pengisian maupun solenoid valve pembuangan.

B. Saran

Penelitian yang dilakukan oleh penulis ini tentunya tidak lepas dari kekurangan dan kelemahan. Oleh karena itu, untuk pengembangannya sistem lebih lanjut diperlukan perhatian terhadap beberapa hal, diantaranya:

1. Alat dapat berjalan dengan sempurna terutama pada bagian solenoid valve jika sumber air berasal dari tandon yang besar, sehingga tekanannya air akan menjadi lebih tinggi.

DAFTAR PUSTAKA

- Hendrawan, Yusuf. 2015. *Rancang Bangun Prototype Alat Penyiram Otomatis Dengan Sistem Timer RTC DS1307 Berbasis Mikrokontroler Atmega 16 Pada Tanaman Aeroponik*. Fakultas Teknologi Pertanian. Universitas Brawijaya. Malang.
- Hermawantyo, Basuko. 2010. *Sistem Pemberian Pupuk Tanaman Secara Otomatis Berbasis Mikrokontroler*. Fakultas Teknologi Industri. Universitas Pembangunan Nasional. Jawa Timur. Surabaya.
- Istivania, Tria. 2014. *Pengaruh Penggunaan Nutrisi Tanaman Dengan Menggunakan Media Yang Berbeda Pada Tanaman Packhoi (Brassica Juncea L.) Hidroponik*. Fakultas Pertanian. Universitas Padjajaran. Jawa Barat. Bandung. Jatinangor.
- Jogiyanto. HM, MBA., 2006. *Konsep Dasar Pemrograman Bahasa C*. Yogyakarta : Andi.
- Nurchahyo, Sidik. 2012. *Aplikasi dan Teknik Pemrograman Mikrokontroler AVR Atmel*. Yogyakarta : Andi Offset.
- Pitowarno, Endra. 2006. *Robotika Desain, Kontrol dan Kecerdasan Buatan*. Yogyakarta: Andi Offset.
- Putra, Agfianto Eko. 2010. *Tip dan Trik Mikrokontroler AT 89 dan AVR*. Yogyakarta : Gava Media.
- Setiawan, Sulhan. 2008. *Mudah dan Menyenangkan Belajar Mikrokontroler*. Yogyakarta : Andi.
- Zulkarnain, Iskandar. 2014. *Rancang Bangun Sistem Hidroponik Pasang Surut Otomatis Untuk Budidaya Tanaman Cabai*. Fakultas Pertanian. Universitas Lampung. Lampung.

LAMPIRAN

PROGRAM CODE VISION AVR

```
/******
```

This program was produced by the

CodeWizardAVR V2.05.3 Standard

Automatic Program Generator

© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 16/11/2015

Author : Bayu Agus Prasetya

Company : Skripsi

Comments:

Chip type : ATmega16

Program type : Application

AVR Core Clock frequency: 12,000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

```
*****/
```

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

```
// I2C Bus functions

#include <i2c.h>

// DS1307 Real Time Clock functions

#include <ds1307.h>

#include <stdio.h>

#include <delay.h>

// Alphanumeric LCD functions

#include <alcd.h>

// Declare your global variables here

unsigned char jam,menit,detik,week,tanggal,bulan,tahun;

unsigned char buffer[16];

/*=====*/

void clock()

{

    rtc_get_time(&jam,&menit,&detik);

    rtc_get_date(&week,&tanggal,&bulan,&tahun);

    lcd_gotoxy(0,0);

    sprintf(buffer,"CLOCK %02u:%02u:%02u",jam,menit,detik);

    lcd_puts(buffer);

    /*lcd_gotoxy(0,1);

    sprintf(buffer,"DATE %02u:%02u:%02u",tanggal,bulan,tahun);
```

```
    lcd_puts(buffer);*/  
}  
  
/*=====*/  
  
void ngisi()  
{  
    lcd_gotoxy(7,1);  
    lcd_putsf("MENGISI");  
}  
  
void nol()  
{  
    lcd_gotoxy(7,1);  
    lcd_putsf("-----");  
    delay_ms(1000);  
    lcd_gotoxy(7,1);  
    lcd_putsf("    ");  
  
}  
  
void buang()  
{  
    lcd_gotoxy(7,1);  
    lcd_putsf("*BUANG*");  
}  
  
void statusair()
```

```
{  
    if(PINA.7 == 1)  
    {  
        lcd_gotoxy(0,1);  
        lcd_putsf("AIR=Y");  
    }  
    else  
    {  
        lcd_gotoxy(0,1);  
        lcd_putsf("AIR=N");  
    }  
}  
  
/*=====*/  
  
void siklus()  
{  
    if(jam == 9 && menit == 59 && detik == 00)  
    {  
        PORTC.0=0;  
        PORTC.1=1;  
        ngisi();  
    }  
    if(PINA.7 == 1)  
    {
```

```
PORTC.0=1;

PORTC.0=1;

nol();

}

{

if(jam == 10 && menit == 9 && detik == 00)

{

PORTC.0=1;

PORTC.1=0;

buang();

}

if(jam == 10 && menit == 14 && detik == 00)

{

PORTC.0=1;

PORTC.1=1;

nol();

}}

}

/*=====*/

void main(void)

{

// Declare your local variables here
```

```
// Input/Output Ports initialization

// Port A initialization

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

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTA=0x00;

DDRA=0x00;

// Port B initialization

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

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTB=0x00;

DDRB=0x00;

// Port C initialization

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

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTC=0xFF;

DDRC=0xFF;
```

```
// Port D initialization

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

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTD=0x00;

DDRD=0x00;

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0=0x00;

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: Timer1 Stopped

// Mode: Normal top=0xFFFF

// OC1A output: Discon.

// OC1B output: Discon.

// 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=0x00;
TCCR1B=0x00;
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=0x00;
TCCR2=0x00;
```

```
TCNT2=0x00;
```

```
OCR2=0x00;
```

```
// External Interrupt(s) initialization
```

```
// INT0: Off
```

```
// INT1: Off
```

```
// INT2: Off
```

```
MCUCR=0x00;
```

```
MCUCSR=0x00;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=0x00;
```

```
// USART initialization
```

```
// USART disabled
```

```
UCSRB=0x00;
```

```
// 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;

// I2C Bus initialization

// I2C Port: PORTB

// I2C SDA bit: 1

// I2C SCL bit: 0

// Bit Rate: 100 kHz

// Note: I2C settings are specified in the
// Project|Configure|C Compiler|Libraries|I2C menu.

i2c_init();

// DS1307 Real Time Clock initialization

// Square wave output on pin SQW/OUT: Off

// SQW/OUT pin state: 0
```

```
rtc_init(0,0,0);

;

// Alphanumeric LCD initialization

// Connections are specified in the

// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:

// RS - PORTD Bit 0

// RD - PORTD Bit 1

// EN - PORTD Bit 2

// D4 - PORTD Bit 4

// D5 - PORTD Bit 5

// D6 - PORTD Bit 6

// D7 - PORTD Bit 7

// Characters/line: 16

lcd_init(16);

//rtc_set_time(12,25,00);

//rtc_set_date(01,16,11,15);

while (1)

    {

        clock();

        statusair();

        siklus();

    }

}
```


PROGRAM VISUAL BASIC

```
Private Sub ComboBaud_Change()  
  
SaveSetting "Evert", "Input", "CommBaud", ComboBaud  
  
End Sub  
  
Private Sub ComboComport_Change()  
  
SaveSetting "Evert", "Input", "Commport", ComboComport  
  
End Sub  
  
Private Sub Command1_Click()  
  
If MSComm2.PortOpen = False Then  
  
    MSComm2.CommPort = ComboComport  
  
    MSComm2.Settings = ComboBaud + ",n,8,1"  
  
    MSComm2.PortOpen = True  
  
  
    MSComm2.InputLen = 0  
  
    MSComm2.RThreshold = 1  
  
  
PortisOpen = True  
  
    Shape9.FillColor = vbGreen  
  
    Command1.Caption = "Disconnect"
```

Else

 MSComm2.PortOpen = False

 Shape9.FillColor = vbWhite

 Command1.Caption = "Connect"

PortisOpen = False

End If

End Sub

Private Sub Form_Load()

 MSComm2.Settings = "9600,N,8,1" 'contoh setting serial port

 MSComm2.InputLen = 0

 MSComm2.RThreshold = 1

 ComboComport = GetSetting("Evert", "Input", "Commport", 1)

 ComboBaud = GetSetting("Evert", "Input", "CommBaud", 9600)

End Sub

Private Sub Frame1_DragDrop(Source As Control, X As Single, Y As Single)

End Sub

Private Sub MSComm2_OnComm()

```
Select Case MSComm2.CommEvent  
  
' Errors  
  
    Case comEventRxParity ' Parity Error.  
  
    MsgBox "Parity"  
  
' Events  
  
    Case comEvReceive ' Received RThreshold # of chars.  
  
        Text1.Text = MSComm2.Input  
  
End Select
```

```
End Sub
```

```
Private Sub Text1_Change()
```

```
Select Case Trim(Text1.Text)  
  
    Case "A"  
  
        Shape1.FillColor = vbRed  
  
        labelketerangan1.Caption = "Pembuangan"  
  
    Case "B"
```

```
Shape1.FillColor = vbGreen
```

```
labelketerangan1.Caption = "Penyiraman"
```

```
Case "a"
```

```
Shape1.FillColor = vbYellow
```

```
labelketerangan1.Caption = "----"
```

```
End Select
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
U = MSComm2.Input
```

```
End Sub
```

PROGRAM ARDUINO UNO

```
//deklarasi sensor

intpenyiraman=10;

intpembuangan=11;

// deklarasi kondisi awal sensor

intkondisisiram = 0;

intkondisibuang = 0;

void setup()

{

Serial.begin(9600);

pinMode(penyiraman,INPUT);

pinMode(pembuangan,INPUT);

}

void loop()

{

kondisisiram = digitalRead(penyiraman);

kondisibuang = digitalRead(pembuangan);

if (kondisisiram == HIGH)

{

Serial.print("A");

}

else if (kondisibuang == HIGH)

{
```

```
Serial.print("B");  
  }  
else  
  {  
Serial.print("C");  
  }  
}
```

**RANCANG BANGUN DAN IMPLEMENTASI SISTEM OTOMATISASI
PENYIRAMAN TANAMAN HIDROPONIK MENGGUNAKAN
SOLENOID VALVE BERBASIS MIKROKONTROLER
AT MEGA 16**

Identitas Responden

Nama :

Pekerjaan :

Berikan tanda cek list (\checkmark) pada pilihan jawaban menurut anda.

No	Pertanyaan	Pilihan		
		B	C	K
1	Tampilan alat keseluruhan ?			
2	Kemudahan penggunaan alat ?			
3	Kelengkapan informasi alat ?			
4	Kemudahan dalam menjalankan alat ?			
5	Kinerja keseluruhan alat ?			

Keterangan : Yogyakarta, Januari 2016

B : Baik

C : Cukup

K : Kurang

(.....)

Saran Terhadap Sistem