DAFTAR PUSTAKA

Bejo, Agus.2008. *C & AVR Rahasia Kemudahan Bahasa C Dalam Mikrokontroler ATMega 8535.*Yogyakarta:Graha Ilmu.

Jamilah. *Pengenalan Bahasa C.* http://jamilah.staff.gunadarma.ac.id/Downloads/files/33355/pengenalan-c-sdcc.pdf. Diakses tanggal 07 April 2015.

Kusumadewi, Sri dan Purnomo, Hari. 2010. *Aplikasi Logika Fuzzy Untuk Pendukung Keputusan.*Yogyakarta:Graha Ilmu.

Lazuardi, Rizky.2011. *Sistem kendali remote control pada robot pengepel dan pengering lantai berbasis mikrokontroler avr atmega 8535*.Palembang : Polsri. Diakses tanggal 05 April 2015.

Novitayantika, Ririn. 2010. *Sistem Pengendali Pendingin Ruangan Berbasis Mikrokontroler ATMega8535.* http://digilib.polsri.ac.id. Diakses tanggal 05 Agustus 2015.

Rabbani, Saddam. 2013. *Navigasi Robot Pemadam Api dengan Metode Fuzzy Logic*.Palembang:Polsri. Diakses tanggal 20 Juli 2015.

Setya, Delta Agus. *Sensor Ultrasonic Sebagai Alat Navigasi Robot Pada Robot Pemadam Api Berbasiskan Mikrokontroler ATMega 8535*. http://eprints .undip.ac.id/20399/1/fix.pdf. Diakses tanggal 05 April 2015.

Sitophila, Monilia, dkk. *Rancang Bangun Atap Sirip Otomatis Menggunakan LDR dan Sensor Tetes Air Hujan Berbasis Mikrokontroler*. Universitas Negeri Malang. http://jurnal-online.um.ac.id/data/artikel/artikel4417FA1491D63077CF5E65C9643822F3.pdf. Diakses tanggal 07 April 2015.

Wahyudi, Ahmad. 2011. Pemanfaatan Media Informasi Pendaftaran Peserta Kursus pada Lembaga Pendidikan Kursus Sinergi Indonesia Berbasis Web. Universitas Islam Attahiriyah : Jakarta.

Wisaksono, Adi. 2011. *Miniatur Pengaman Jemuran Otomatis Berbasis Mikrokontroler ATMEGA 8535*. Politeknik Negeri Semarang : Semarang.

Yagusandri, Ariel. 2011. *Rancang Bangun Prototipe Sistem Aktuator Sirip Roket Menggunakan Motor Servo*. Universitas Indonesia : Depok.

LAMPIRAN

LISTING PROGRAM

```
/*****************************************************
                PROGRAM FUZZY SENSOR CAHAYA
*****************************************************/

#include <mega8535.h>
#include <stdio.h>
//
//DEKLARASI VARIABEL
unsigned char data1,h1,h2,h3,h4;
unsigned char d2,d3,d5;
int a1, b1,c1,d1,x1,x5,z1,z2,cahaya1;
int a2,b2;
int data_naik, data_turun,Zk1,Zk3,Zk;
int k1,derajat1;
#asm
   .equ __lcd_port=0x18 ;PORTB
#endasm
#include <lcd.h>
#include <delay.h>
#define ADC_VREF_TYPE 0x20
char lcd_buffer[33];
char lcd_buf1[17];
char lcd_buf2[17];
// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Delay needed for the stabilization of the ADC input voltage
```

```
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCH;
}

// DEKLARASI SUBRUTINE
void ambil_data_cahaya();
void ambil_data_hujan();
void proses();
void rule();
void hit1();
void tampil();
void tampil2();
void tampil3();
void buka();
void buka_atap();
void tutup();
void cek_posisi();
void cek_posisi2();
void cek_posisi3();
void buka12a();
void tutup12a();
void cek_hujan1();
void cek_hujan2();
void cek_hujan3();
void tutup12b();
```

```
void buka12b();

void cek_lagi();



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=0x00;

DDRC=0xff;
```

```
// Port D 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

PORTD=0x00;

DDRD=0xFF;


// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=FFh

// OC0 output: Disconnected

TCCR0=0x00;

TCNT0=0x00;

OCR0=0x00;


// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: Timer 1 Stopped

// Mode: Normal top=FFFFh

// OC1A output: Discon.

// OC1B output: Discon.

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer 1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=0xA1;

TCCR1B=0x0B;
```

```
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: Timer 2 Stopped
// Mode: Normal top=FFh
// 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;

// Analog Comparator initialization
```

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

// ADC initialization
// ADC Clock frequency: 1000.000 kHz
// ADC Voltage Reference: AREF pin
// ADC High Speed Mode: Off
// ADC Auto Trigger Source: None
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;
SFIOR&=0xEF;

// LCD module initialization
lcd_init(16);
//PROGRAM UTAMA
while (1){
//   buka();
//   delay_ms(1000);
//   tutup();
//   delay_ms(1000);
h1=read_adc(0);
h2=read_adc(1);
h3=read_adc(2);
h4=read_adc(3);
data1=~read_adc(4);         // sensor cahaya
// tampil();
```

```
//  ambil_data_hujan();
ambil_data_cahaya();
// tampil();
hit1();                    // proses fuzzyfikasi sensor cahaya
//tampil2();
proses();                  // proses defuzzyfikasi
tampil3();
// cek_posisi();
rule();                   // proses atur kipas  */
};} void tampil(){
while(1){
lcd_clear();
lcd_gotoxy(0,0);
sprintf(lcd_buf1,"h1=%u,h2=%u",h1,h2);
lcd_puts(lcd_buf1);
lcd_gotoxy(0,1);
sprintf(lcd_buf1,"h3=%u,h4=%u",h3,h4);
lcd_puts(lcd_buf1);
delay_ms(100);
break;}}

// ambil data sensor hujan
void ambil_data_hujan(){
while(1){
if ((h1>=100) || (h2>=100) || (h3>=100) || (h4>=100)){
tutup();
PORTC.0=0;                 // kipas OFF
buka_atap();
break;}
PORTC.0=0;
```

```
break;}}
// buka
void buka_atap(){
while(1){
h1=read_adc(0);
h2=read_adc(1);
h3=read_adc(2);
h4=read_adc(3);
// tampil();
if ((h1<=100) && (h2<=100) && (h3<=100) && (h4<=100)){
buka();
PORTC.0=0;            // KIPAS OFF
break;}}}
// Ambil data sensor cahaya
void ambil_data_cahaya(){
while(1){
cahaya1=data1;//~read_adc(4);
x1=cahaya1;
break;  }}
// hitung fuzzy sensor cahaya
void hit1(){
while(1){
a1=30;                  // batas GELAP
b1=70;                  // batas MENDUNG
c1=120;                  // BATAS TERANG
x1=cahaya1;               // data  pembacaan Sensor
x5=cahaya1;                // konstanta cahaya

if (x1>=30 && x1<70)    {
// fungsi keanggotaan GELAP
```

```
z1=(b1-x1)*100;

z2=b1-a1;

data_turun=(z1/z2);

// fungsi keanggotaan MENDUNG

z1=(x1-a1)*100;

z2=b1-a1;

data_naik=(z1/z2);   }

if (x1>=70 && x1<120) {

// fungsi keanggotaan MENDUNG

z1=(c1-x1)*100;

z2=c1-b1;

data_turun=(z1/z2);

// fungsi keanggotaan TERANG

z1=(x1-b1)*100;

z2=c1-b1;

data_naik=(z1/z2); }

// Menentukan nilai maksimal (ambil nilai derajat keanggotaan
terbesar)

if (data_naik>data_turun )

{derajat1=data_naik;}

if (data_turun>data_naik)

{derajat1=data_turun;}

break;}}

void tampil2(){

while(1){

lcd_clear();

 lcd_gotoxy(0,0);

 sprintf(lcd_buf1,"suhu=%u,x=%u",cahaya1,data_turun);

 lcd_puts(lcd_buf1);

 lcd_gotoxy(0,1);

 sprintf(lcd_buf1,"y=%u,d=%u",data_naik,derajat1);
```

```
 lcd_puts(lcd_buf1);
 delay_ms(100);
 break;}}

// proses defuzzyfikasi
void proses(){
while(1){
Zk1=derajat1*x5;
Zk3=Zk1;
k1=derajat1;
Zk= Zk3/k1;
break;}}
void tampil3(){
while(1){
lcd_clear();
lcd_gotoxy(0,0);
sprintf(lcd_buf1,"cahaya=%u,k1=%u",cahaya1,k1);
lcd_puts(lcd_buf1);
lcd_gotoxy(0,1);
sprintf(lcd_buf1,"Zk3=%u,Zk=%u",Zk3,Zk);
lcd_puts(lcd_buf1);
delay_ms(100);
break;}}
// atur kipas
void rule(){
while(1){
if (Zk<=50)              // tutup full
{ d1+=1;
PORTC.1=1;
cek_posisi();
```

```
break;}
if (Zk>50 && Zk<=95)   // tutup 1/2
{ d2+=1;
PORTC.1=1.;
cek_posisi2();
break;}
if (Zk>95)              // buka full
{ d3+=1;
PORTC.1=0.;
cek_posisi3();
break;}
//break;}}
// cek posisi awal atap
void cek_posisi(){
while(1){
if (d2>=1)
{
tutup12b();           // tutup 1/2 gelap
d2=0;
// delay_ms(3000);
cek_hujan1();
break;}
if (d3>=1)
{
tutup();
d3=0;
// delay_ms(3000);
cek_hujan1();
break;}
if (d1==1){
```

```
tutup();
// delay_ms(3000);
cek_hujan1();
 break;}
 break;
 }
 }
 // posisi ke 2
 void cek_posisi2(){
 while(1) {
 if (d1>=1){
 buka12b();     // buka 1/2 gelap
 d1=0;
 // delay_ms(3000);
 cek_hujan2();
 break;}
 if (d3>=1) {
 tutup12a();    // tutup 1/2 mendung
 d3=0;
 delay_ms(3000);
 cek_hujan2();
 break;}
 if (d2==1){
 tutup12a();
 // delay_ms(3000);
 cek_hujan2();
 break;}
 break;}}
 //posisi ke 3
 void cek_posisi3(){
```

```
while(1){
if (d1>=1){
buka();     // buka full
d1=0;
// delay_ms(3000);
cek_hujan3();
 break;}
 if (d2>=1){
 buka12a();
 d2=0;
 // delay_ms(3000);
 cek_hujan3();
 break;}
 if (d3==1){
 //buka();
 // delay_ms(3000);
 cek_hujan3();
 break; }
 break; }}
 // cek apakah hujan turun
 void cek_hujan1() {
 while(1){
 h1=read_adc(0);
 h2=read_adc(1);
 h3=read_adc(2);
 h4=read_adc(3);
 if ((h1>=100) || (h2>=100) || (h3>=100) || (h4>=100)){
 //tutup14();
 PORTC.1=1;                 // kipas ON
 cek_lagi();
```

```
 break;}
 // buka();
 break; }}
 void cek_hujan2(){
 while(1){
 h1=read_adc(0);
 h2=read_adc(1);
 h3=read_adc(2);
 h4=read_adc(3);
 if ((h1>=100) || (h2>=100) || (h3>=100) || (h4>=100)){
 tutup12b();
 PORTC.1=1;                 // kipas ON
 cek_lagi();
 break;}
 break;}}
//
void cek_hujan3(){
while(1){
h1=read_adc(0);
h2=read_adc(1);
h3=read_adc(2);
h4=read_adc(3);
if ((h1>=100) || (h2>=100) || (h3>=100) || (h4>=100)){
tutup();
PORTC.1=1;                 // kipas ON
cek_lagi();
break;}
break;}}
void cek_lagi(){
while(1){
```

```
h1=read_adc(0);
h2=read_adc(1);
h3=read_adc(2);
h4=read_adc(3);
if ((h1<100) && (h2<100) && (h3<100) && (h4<100)){
PORTC.1=1;                // kipas ON
buka();                   // jika tidak, maka buka full
// delay_ms(2000);
break;}}}
// tutup atap full
void tutup(){
while(1){
int i;
for (i=0; i<284; i++){
PORTC=1;          //memberikan pulsa high
delay_ms(0.9);      //pulsa high diberikan selama 0.9 ms
PORTC=0;
delay_ms(20);}
break; }}
// buka atap full
void buka(){
while(1){
int i;
for (i=0; i<284; i++){
PORTC.0=1;          //memberikan pulsa high
delay_ms(2.1);      //pulsa high diberikan selama 2.1 ms
PORTC.0=0;
delay_ms(20);}
break; }}
void tutup12a()  // tutup atap 1/2 kondisi mendung {
```

```
while(1){
int i;
for (i=0; i<142; i++){
PORTC=1;          //memberikan pulsa high
delay_ms(0.9);      //pulsa high diberikan selama 0.9 ms
PORTC=0;
delay_ms(20);}
break;}}
// buka atap 1/2 kondisi mendung
void buka12a(){
while(1){
int i;
for (i=0; i<147; i++){
PORTC.0=1;          //memberikan pulsa high
delay_ms(2.1);      //pulsa high diberikan selama 2.1 ms
  PORTC.0=0;
  delay_ms(20);}
  break; }}
  // tutup atap 1/2 kondisi gelap
  void tutup12b(){
  while(1){
  int i;
  for (i=0; i<142; i++){
  PORTC=1;          //memberikan pulsa high
  delay_ms(0.9);      //pulsa high diberikan selama 0.9 ms
  PORTC=0;
  delay_ms(20);}
  break; }}
  // buka atap 1/2 kondisi gelap
  void buka12b(){
```

```
while(1){
int i;
for (i=0; i<145; i++){
PORTC.0=1;          //memberikan pulsa high
delay_ms(2.1);      //pulsa high diberikan selama 2.1 ms
PORTC.0=0;
delay_ms(20);}
break;}}
```