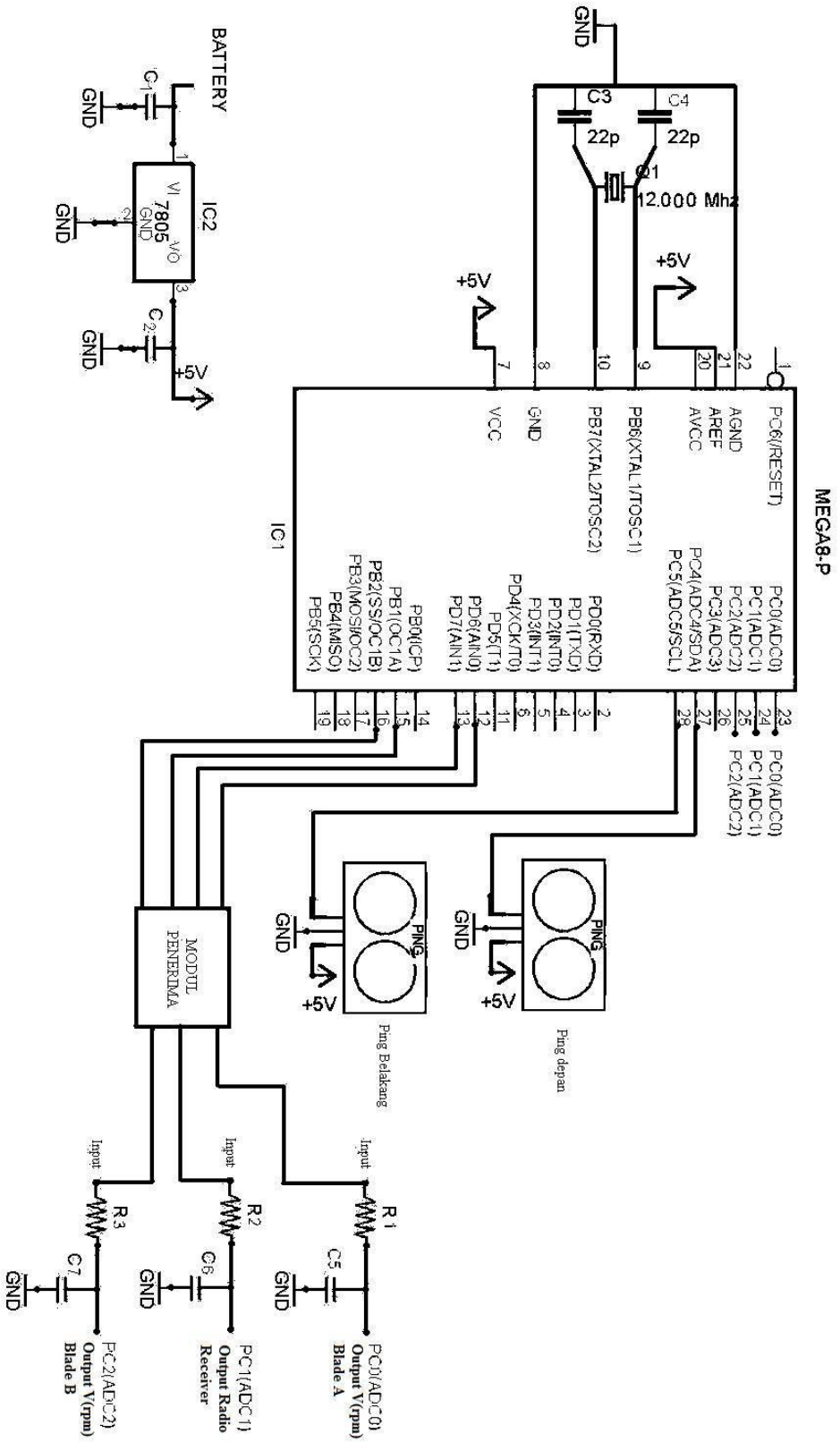
LAMPIRAN A

Gambar Rangkaian Elektronika



LAMPIRAN B

Program Pada Mikrokontroler

```
/*****************************************************
Project : Unmanned Helicopter Automatic Landing
Version : ATMega
Date    : 01/11/2011
Author  : Freeware, for evaluation and non-commercial use only
Company : S1 Fisika Instrument FMIPA USU
Comments:


Chip type           : ATmega8
Program type        : Application
Clock frequency     : 12,000000 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*****************************************************/

#include <mega8.h>
#include <delay.h>

#define ekora PORTD.6=0
#define ekorb PORTD.7=0

#define pulse1 PORTC.4 // untuk sensor ping ultrasonik
#define echo1 PINC.4
#define arah1 DDRC.4
#define out1 1
#define inp1 0
#define pulse2 PORTC.5
```

```
#define echo2 PINC.5
#define arah2 DDRC.5
#define out2 1
#define inp2 0


unsigned int count1=0;
unsigned int count2=0;
unsigned int radio;
unsigned char last_1;
unsigned char last_2;
unsigned char sensor_1;
unsigned char sensor_2;

#define ADC_VREF_TYPE 0xC0

// Read the AD conversion result
unsigned int 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 ADCW;
}

// Declare your global variables here

void jony(void)
```

```
{// untuk sensor ping ultrasonik a
 count1=0;
 arah1=out1;
 pulse1=1;
 delay_us(5);
 pulse1=0;
 arah1=inp1;
 pulse1=1;
 while (echo1==1 )
 {
 count1++;
 delay_us(2);
 };
 sensor_1=count1*0.034446;



}

void smith(void)
{// untuk sensor ping ultrasonik b
count2=0;
arah2=out2;
pulse2=1;
delay_us(5);
pulse2=0;
arah2=inp2;
pulse2=1;
while (echo2==1 )
 {
 count2++;
 delay_us(2);
 };
 sensor_2=count2*0.034446;
```

```
}


void sinambela(void)
        { // untuk input adc dan output motor

         radio=read_adc(1);
         last_1=read_adc(0);
         last_1=read_adc(2);



         OCR1AL=last_1;
         OCR1BL=last_2;



         while (radio<=90)
            {
           if(OCR1AL>0)
           OCR1AL--;
           delay_ms(5);

           if(OCR1BL>0)
           OCR1BL--;
           delay_ms(5);

           jony();
           smith();

           if(sensor_1%sensor_2>=10)
             {
             PORTD.7=0;
```

```
            PORTD.6=1;
            OCR1AL--;
            delay_us(200);
            }

          if(sensor_2%sensor_1>=10)
            {
            PORTD.6=0;
            PORTD.7=1;
            OCR1BL--;
            delay_us(200);
            }

          };


        }

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

// Input/Output Ports initialization
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=Out Func1=Out Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=0 State1=0 State0=T
PORTB=0x00;
DDRB=0x06;

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTC=0x00;
DDRC=0x00;

// 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
TCCR0=0x00;
TCNT0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Fast PWM top=00FFh
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// 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=0x08;
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
MCUCR=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: 93,750 kHz
// ADC Voltage Reference: Int., cap. on AREF
ADMUX=ADC_VREF_TYPE & 0xff;
```

```
ADCSRA=0x87;



while (1)
    {
    // Place your code here

    jony();
    smith();
    sinambela();

    };
}
```