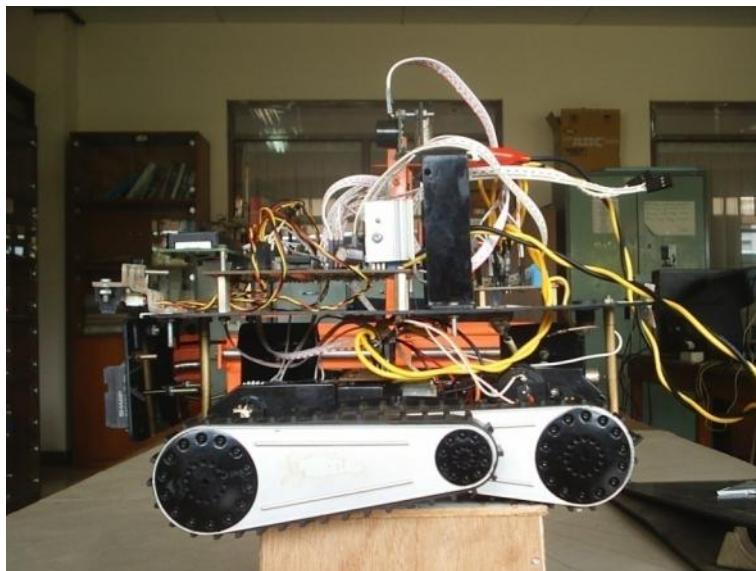


LAMPIRAN A
FOTO ROBOT MOBIL

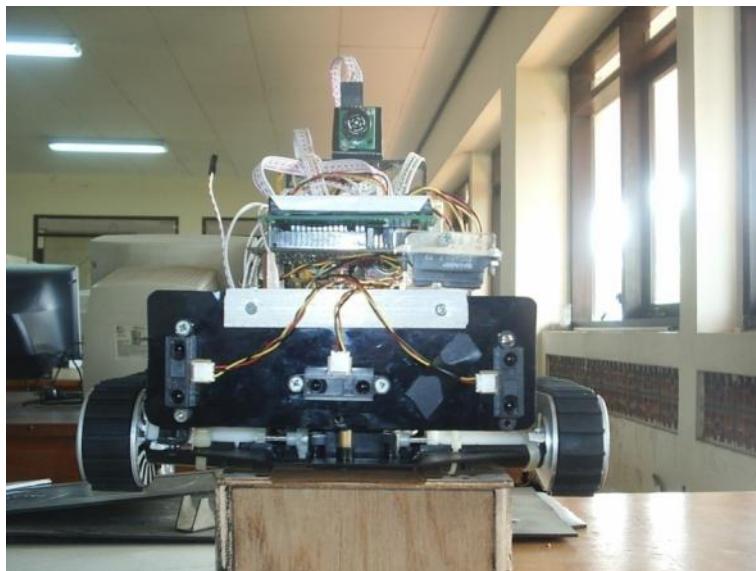
Tampak Samping



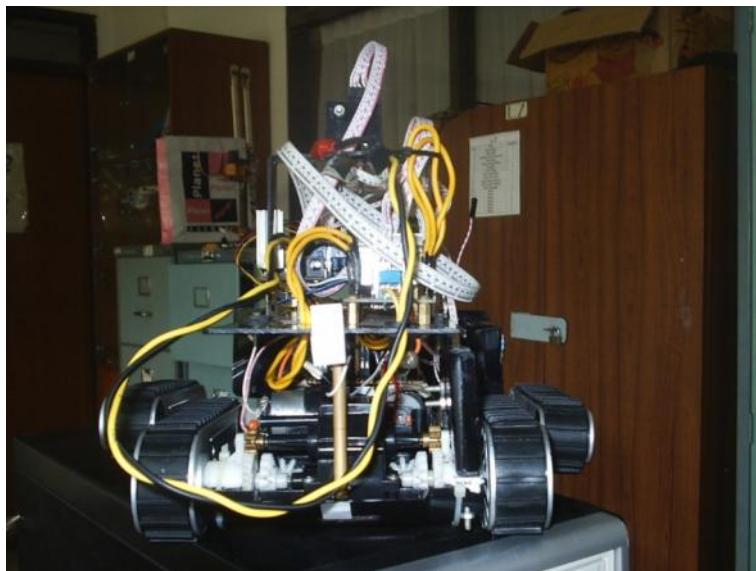
Tampak Atas



Tampak Depan



Tampak Belakang



LAMPIRAN B
PROGRAM PADA PENGONTROL
ATMEGA16

PROGRAM UTAMA

```
*****
```

This program was produced by the
CodeWizardAVR V1.25.3 Professional
Automatic Program Generator
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project :
Version :
Date : 1/13/2009
Author : Lab Instrumentasi
Company : UKM
Comments:

Chip type : ATmega16
Program type : Application
Clock frequency : 11.059200 MHz
Memory model : Small
External SRAM size : 0
Data Stack size : 256

```
#include <mega16.h>
#include <stdio.h>
#include <delay.h>
#include <math.h>
```

```
// Declare your global variables here
bit a,b,c,d,e,chy;
char text[16],rt,temp;
char konter1,konter2;
unsigned char texti2c[16];
char textr[16],textl[16],textt[16],texte[16];
int dr,dl,dt,edde;
int sr,sl,st,edge;
unsigned int p,bil1,bil2;
```

```
//fungsi-fungsi
void init_system(void);
void angkat_tuas_belakang_305(void);
void angkat_tuas_belakang_260(void);
void angkat_tuas_depan_180(void);
void angkat_tuas_depan_260(void);
void turun_tuas_belakang_260(void);
void turun_belakang_45(void);
```

```

void turun_tuas_belakang_260(void);
void turun_tuas_depan_180(void);
void turun_tuas_depan_260(void);
void maju(void);
void maju_1(void);
void stop(void);
void rotasi_depan(char input);
void rotasi_belakang(char input);
void baca_sensor(void);
void baca_i2c(void);
void angkat_tuas_depan_30(void);
void angkat_tuas_belakang_180(void);
void turun_tuas_belakang_180(void);
void edge_detection(void);

// I2C Bus functions
#asm
    .equ __i2c_port=0x18 ;PORTB
    .equ __sda_bit=6
    .equ __scl_bit=7
#endifasm
#include <i2c.h>

// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x15 ;PORTC
#endifasm
#include <lcd.h>

#define ADC_VREF_TYPE 0x60

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

void main(void)
{

```

```

init_system();

while (1)
{
    awal:
    baca_sensor();
    delay_ms(10);

    if(dr >=20 && dl >=20 && dt >=20)
    {
        maju_1();
    }

    if(dr <=17 && dl <=17 && dt <=17)
    {
        lcd_clear();
        lcd_putsf("kanan kiri tengah");
        stop();
        lcd_clear();
        //algoritma naik tangga
        for(temp=0;temp<10;temp++)
        {
            baca_i2c();
            delay_ms(200);
        }

        if(p<=58 && p>=40)
        {
            lcd_clear();
            turun_belakang_45();
            delay_ms(500);
            maju_1();
            delay_ms(650);
            stop();
            angkat_tuas_depan_180();
            delay_ms(1000);
            angkat_tuas_belakang_305();
            delay_ms(1000);
            maju_1();
            delay_ms(450);
            stop();
            turun_tuas_belakang_260();
            delay_ms(1000);
            turun_tuas_depan_180();
            delay_ms(1000);
        }
    }

    //algoritma lewati benda
    if(dr >=20 && dl >=20 && dt <=20)
    {
        for(temp=0;temp<10;temp++)
    }
}

```

```

{
baca_i2c();
delay_ms(200);
}

if(p>=65)
{
lcd_clear();
lcd_putsf("sensor tengah ajah");
stop();
lcd_clear();
angkat_tuas_depan_260();
delay_ms(1000);
angkat_tuas_belakang_260();
delay_ms(1000);
maju_1();
delay_ms(1100);
stop();
lcd_clear();
turun_tuas_belakang_260();
delay_ms(1000);
turun_tuas_depan_260();
delay_ms(1000);
goto awal;
}

if(dr <=17 && dl >=30 && dt >=30)
{
lcd_clear();
lcd_putsf("sensor kanan sajah");
goto again;
}

if(dr >=30 && dl <=17 && dt >=30)
{
lcd_clear();
lcd_putsf("sensor kiri sajah");
goto again;
}

if(dr >=30 && dl <=17 && dt <=17)
{
lcd_clear();
lcd_putsf("sensor tengah dan kiri");
goto again;
}

if(dr <=17 && dl >=30 && dt <=17)
{
lcd_clear();
lcd_putsf("sensor kanan dan tengah");
}

```

```

        goto again;
    }

//algoritma naiki dan turuni benda
if(dr <=17 && dl <=17 && dt >=30)
again:
{
stop();
lcd_clear();
for(temp=0;temp<10;temp++)
{
baca_i2c();
delay_ms(200);
}

if(p>=65)

{
maju_1();
delay_ms(1000);
angkat_tuas_depan_180();
delay_ms(800);
angkat_tuas_belakang_260();
delay_ms(800);
maju_1();
delay_ms(550);
stop();
delay_ms(800);
turun_tuas_belakang_260();
delay_ms(400);
lcd_clear();

for(temp=0;temp<25;temp++)
{
edge_detection();
delay_ms(75);
}

if(edde <=14)
{
lcd_clear();
lcd_putsf("masih jauh");
delay_ms(1000);
while(edde <=14)
{
maju_1();
edge_detection();
}
goto turun;
}

if(edde >=15)
turun:

```

```

    {
        lcd_clear();
        lcd_putsf("sudah diujung");
        stop();
        maju_1();
        delay_ms(250);
        stop();
        delay_ms(800);
        angkat_tuas_depan_30();
        delay_ms(800);
        angkat_tuas_belakang_180();
        delay_ms(800);
        maju_1();
        delay_ms(400);
        stop();
        turun_tuas_depan_260(); //turun_tuas_depan_210 ();
        delay_ms(800);
        turun_tuas_belakang_180();
    }

}

}

}

}

void init_system(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=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=T State6=T State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTB=0x00;
DDRB=0x3F;

// 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=0x00;

// 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: Ph. correct 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=0x00;
// TCNT1H=0x00;
// TCNT1L=0x00;
// ICR1H=0x00;
// ICR1L=0x00;
// OCR1AH=0x00;
// OCR1AL=0x00;
// OCR1BH=0x00;
// OCR1BL=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 172.800 kHz
// Mode: Fast PWM top=ICR1
// 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=0xA2;
TCCR1B=0x1B;

```

```

TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x03;
ICR1L=0xE8;
OCR1AH=0x03;
OCR1AL=0xE8;
OCR1BH=0x03;
OCR1BL=0xE8;

// 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: 691.200 kHz
// ADC Voltage Reference: AVCC pin
// 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;

// I2C Bus initialization
i2c_init();

// LCD module initialization
lcd_init(16);
lcd_putsf("(Robot Tigor) 0422140");
delay_ms(1000);
PORTA.5=1;
PORTA.6=1;
}

```

```

void rotasi_depan(char input)
{
konter1=0;
e=PINA.6;

while(konter1<=input)
{

switch(rt)
{
case 0:a=PINA.6;break;
case 1:b=PINA.6;break;
case 2:c=PINA.6;break;
case 3:d=PINA.6;break;
}
rt++;
if(rt==4)
rt=0;

if(e==1 && a==0 && b==0 && c==0 && d==0)
{konter1++;e=0;}
if(e==0 && a==1 && b==1 && c==1 && d==1)
{konter1++;e=1;}

lcd_gotoxy(0,0);
sprintf(text,"rot dpn=%2u kali",konter1);
lcd_puts(text);

delay_us(1);
}
return;
}

void rotasi_belakang(char input)
{
konter2=0;
b=PINA.5;

while(konter2<=input)
{
if(PINA.5!=b)
{konter2++;
b=PINA.5;}

lcd_gotoxy(0,1);
sprintf(text,"rot blg=%2u kali",konter2);
lcd_puts(text);

delay_ms(1);
}
return;
}

```

```

void maju(void)
{
if(chy==0)
{
PORTD.0=0;
PORTD.1=1;

PORTD.2=0;
PORTD.3=1;
delay_ms(50);
}
chy=1;
OCR1A=620;
OCR1B=470;

if(chy==1)
{
PORTD.0=0;
PORTD.1=1;

PORTD.2=0;
PORTD.3=1;
delay_ms(50);
}
chy=0;
OCR1A=620;
OCR1B=470;
}

void maju_1(void)
{
OCR1A=820;
OCR1B=620;

PORTD.0=0;
PORTD.1=1;

PORTD.2=0;
PORTD.3=1;
}

void stop(void)
{
PORTD.0=1;
PORTD.1=1;

PORTD.2=1;
PORTD.3=1;

delay_ms(1000);
}

```

```

void turun_tuas_depan_260(void)
{
    PORTB.3=0;
    PORTB.4=1;
    PORTB.5=1;

    rotasi_depan(20);

    PORTB.3=1;
    PORTB.4=0;
    PORTB.5=1;
    delay_ms(44);

    PORTB.3=1;
    PORTB.4=1;
    PORTB.5=1;
}

void angkat_tuas_depan_260(void)
{
    PORTB.3=1;
    PORTB.4=0;
    PORTB.5=1;

    rotasi_depan(20);

    PORTB.3=0;
    PORTB.4=1;
    PORTB.5=1;
    delay_ms(44);

    PORTB.3=1;
    PORTB.4=1;
    PORTB.5=1;
}

void angkat_tuas_depan_180(void)
{
    PORTB.3=1;
    PORTB.4=0;
    PORTB.5=1;

    rotasi_depan(14);

    PORTB.3=0;
    PORTB.4=1;
    PORTB.5=1;
    delay_ms(80);

    PORTB.3=1;
    PORTB.4=1;
    PORTB.5=1;
}

```

```
}
```

```
void angkat_tuas_depan_30(void)
{
    PORTB.3=1;
    PORTB.4=0;
    PORTB.5=1;
```

```
    rotasi_depan(5);
```

```
    PORTB.3=0;
    PORTB.4=1;
    PORTB.5=1;
    delay_ms(44);
```

```
    PORTB.3=1;
    PORTB.4=1;
    PORTB.5=1;
}
```

```
void turun_tuas_depan_180(void)
{
```

```
    PORTB.3=0;
    PORTB.4=1;
    PORTB.5=1;
```

```
    rotasi_depan(14);
```

```
    PORTB.3=1;
    PORTB.4=0;
    PORTB.5=1;
    delay_ms(80);
```

```
    PORTB.3=1;
    PORTB.4=1;
    PORTB.5=1;
}
```

```
void angkat_tuas_belakang_260(void)
{
```

```
    PORTB.0=1;
    PORTB.1=0;
    PORTB.2=1;
```

```
    rotasi_belakang(20);
```

```
    PORTB.0=0;
    PORTB.1=1;
```

```

PORTB.2=1;
delay_ms(25);

PORTB.0=1;
PORTB.1=1;
PORTB.2=1;
}

void angkat_tuas_belakang_180(void)
{
    PORTB.0=1;
    PORTB.1=0;
    PORTB.2=1;

    rotasi_belakang(14);

    PORTB.0=0;
    PORTB.1=1;
    PORTB.2=1;
    delay_ms(25);

    PORTB.0=1;
    PORTB.1=1;
    PORTB.2=1;
}

void angkat_tuas_belakang_305(void)
{
    PORTB.0=1;
    PORTB.1=0;
    PORTB.2=1;

    rotasi_belakang(24);

    PORTB.0=0;
    PORTB.1=1;
    PORTB.2=1;
    delay_ms(80);

    PORTB.0=1;
    PORTB.1=1;
    PORTB.2=1;
    delay_ms(1000);
}

void turun_tuas_belakang_260(void)
{
    PORTB.0=0;
    PORTB.1=1;
    PORTB.2=1;

    rotasi_belakang(20);
}

```

```

PORTB.0=1;
PORTB.1=0;
PORTB.2=1;
delay_ms(50);

PORTB.0=1;
PORTB.1=1;
PORTB.2=1;
}

void turun_tuas_belakang_180(void)
{
    PORTB.0=0;
    PORTB.1=1;
    PORTB.2=1;

    rotasi_belakang(14);

    PORTB.0=1;
    PORTB.1=0;
    PORTB.2=1;
    delay_ms(50);

    PORTB.0=1;
    PORTB.1=1;
    PORTB.2=1;
}

void turun_belakang_45(void)
{
    PORTB.0=0;
    PORTB.1=1;
    PORTB.2=1;

    rotasi_belakang(3);

    PORTB.0=1;
    PORTB.1=0;
    PORTB.2=1;
    delay_ms(10);

    PORTB.0=1;
    PORTB.1=1;
    PORTB.2=1;

}

void baca_sensor(void)
{
    sr=read_adc(2);
}

```

```

if(sr<20)
sr=20;
dr=2141.72055 * (pow(sr,-1.078867));

sprintf(textr,"IRR=%3u ", dr);

sl=read_adc(1);
if(sl<20)
sl=20;
dl=2141.72055 * (pow(sl,-1.078867));

sprintf(textl,"IRL=%3u", dl);

st=read_adc(0);
if(st<20)
st=20;
dt=2141.72055 * (pow(st,-1.078867));

sprintf(textt,"IRM=%3u ", dt);

lcd_clear();
lcd_gotoxy(0,0);
lcd_puts(textr);
lcd_gotoxy(9,0);
lcd_puts(textl);
lcd_gotoxy(0,1);
lcd_puts(textt);
}

void baca_i2c(void)
{
i2c_start();
i2c_write(0xE4);
i2c_write(0);
i2c_write(0x51);
i2c_start();
i2c_write(0xE4);
i2c_write(2);
i2c_start();
i2c_write(0xE5);
bil1=i2c_read(1);
bil2=i2c_read(0);
i2c_stop();
p=(bil1*256)+bil2;
lcd_clear();
lcd_gotoxy(0,0);
sprintf(texti2c," i2c= %2ucm", p );
lcd_puts(texti2c);
}

void edge_detection(void)
{
edge=read_adc(3);

```

```
if(edge<20)
edge=20;
edde=2141.72055 * (pow(edge,-1.078867));

sprintf(texte,"IR edge=%3u", edde);
lcd_clear();
lcd_gotoxy(0,0);
lcd_puts(texte);
}
```

LAMPIRAN C
DATASHEET