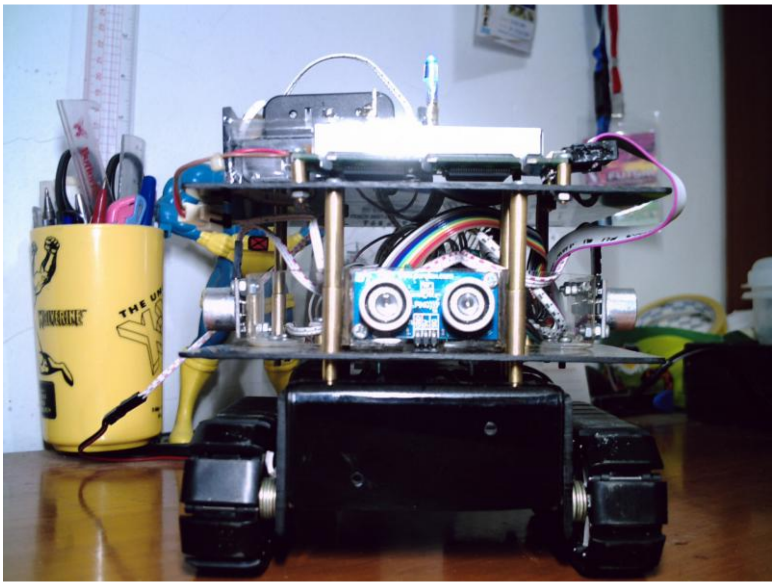
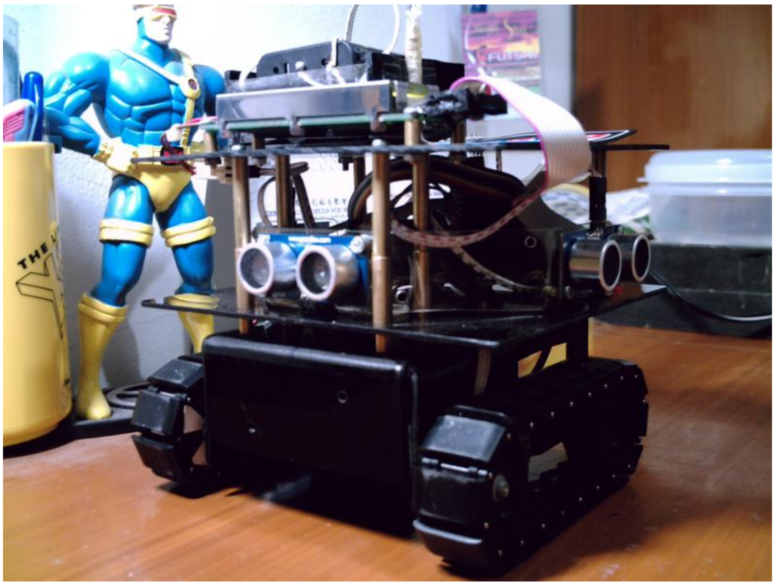
LAMPIRAN A

FOTO ROBOT MOBIL TANK

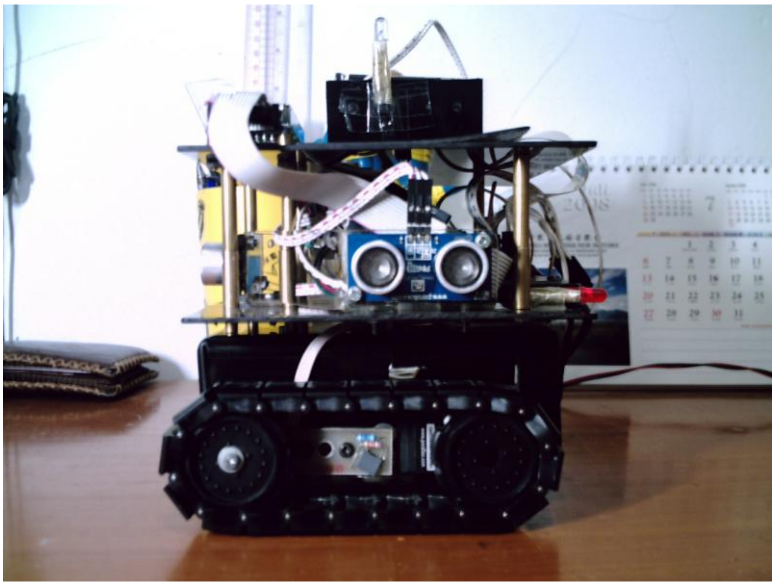
TAMPAK DEPAN



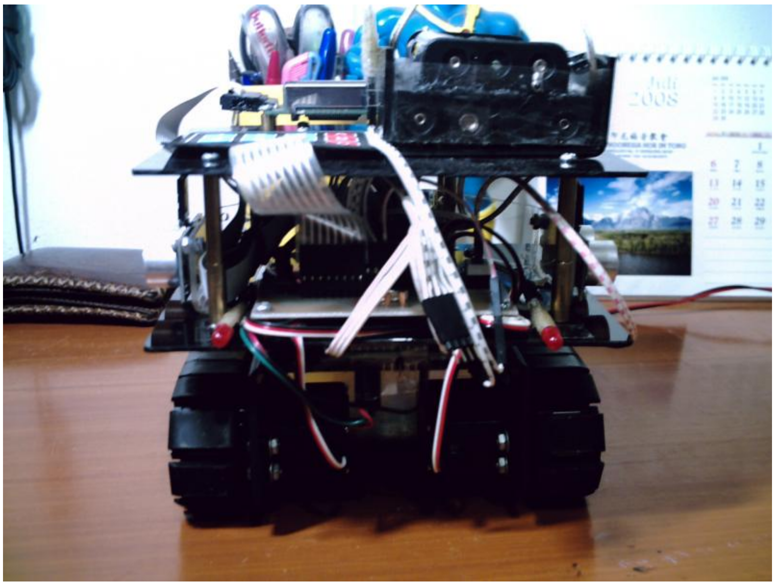
TAMPAK SAMPING DEPAN



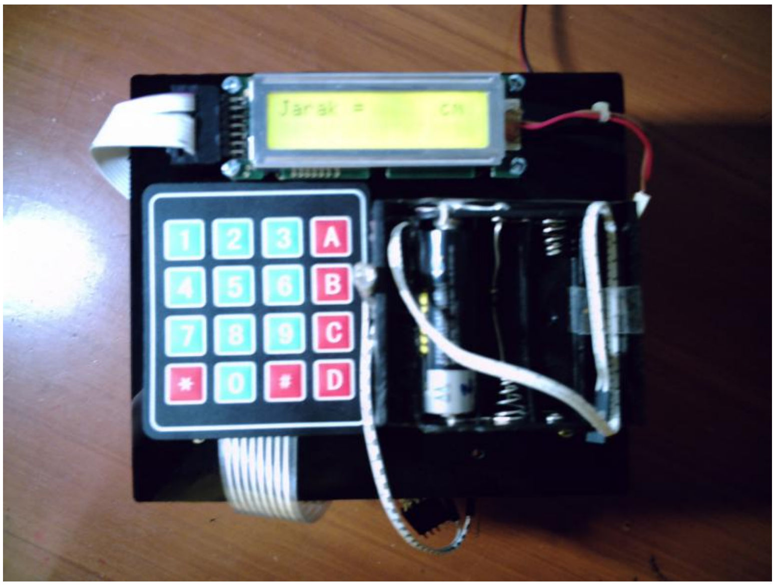
TAMPAK SAMPING



TAMPAK BELAKANG



TAMPAK ATAS



LAMPIRAN B

PROGRAM PADA PENGONTROL MIKRO

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    : 6/24/2008
Author  : Lab
Company : Lab
Comments:


Chip type           : ATmega16
Program type        : Application
Clock frequency     : 12.000000 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 256
*****************************************************/

#include <mega16.h>
#include <delay.h>
#include <stdio.h>
#include <scankeypadB.h>

char eeprom aaaa=0;

// I2C Bus functions
#asm
   .equ __i2c_port=0x1B ;PORTA
   .equ __sda_bit=6
   .equ __scl_bit=7
#endasm
#include <i2c.h>

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

// Declare your global variables here

        int i;
        unsigned int p;
        int apa_kek1,apa_kek2,apa_kek3,apa_kek4;
        unsigned int us0,us1,us2;
        unsigned int ping0,ping1,ping2;
        char lcd_buffer[30];
        int cmps,cmps_0,cmps_1,cmps_target,cmps_error;
        int rotL,rotR;
        int hitung_rotR[5],jumlah,total;
        int x,temp,array[15],input,satuan;

#include <ping.h>
#include <compass.h>
#include <rotary.h>
#include <belok.h>
```

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

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=Out Func3=Out Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=0 State3=0 State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x18;

// 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=0x00;

// Port D initialization
// Func7=In Func6=In Func5=Out Func4=Out Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=0 State4=0 State3=P State2=P State1=T State0=T
PORTD=0x0C;
DDRD=0x30;

// 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: 1500.000 kHz
// Mode: Ph. & fr. cor. 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=0xA0;
TCCR1B=0x12;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x3A;
ICR1L=0x98;
OCR1AH=0x04;
OCR1AL=0x65;
OCR1BH=0x04;
OCR1BL=0x65;

// 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;

// I2C Bus initialization
i2c_init();

// LCD module initialization
lcd_init(16);

rotary_awal();

start:
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Jarak = ");

lcd_gotoxy(13,0);
lcd_putsf("cm");

lcd_gotoxy(8,0);
x=8;

while (1)
      {
      // Place your code here
        awal:
        temp=scan_keypadB();

        switch(temp)
        {
        case 1:
        lcd_gotoxy(x,0);
        sprintf(lcd_buffer,"%d",temp);lcd_puts(lcd_buffer);
        delay_ms(500);
        array[x]=temp;
        x=x+1;
        break;

        case 2:
        lcd_gotoxy(x,0);
        sprintf(lcd_buffer,"%d",temp);lcd_puts(lcd_buffer);
        delay_ms(500);
        array[x]=temp;
        x=x+1;
        break;

        case 3:
        lcd_gotoxy(x,0);
        sprintf(lcd_buffer,"%d",temp);lcd_puts(lcd_buffer);
        delay_ms(500);
```

```
array[x]=temp;
x=x+1;
break;

case 4:
lcd_gotoxy(x,0);
sprintf(lcd_buffer,"%d",temp);lcd_puts(lcd_buffer);
delay_ms(500);
array[x]=temp;
x=x+1;
break;

case 5:
lcd_gotoxy(x,0);
sprintf(lcd_buffer,"%d",temp);lcd_puts(lcd_buffer);
delay_ms(500);
array[x]=temp;
x=x+1;
break;

case 6:
lcd_gotoxy(x,0);
sprintf(lcd_buffer,"%d",temp);lcd_puts(lcd_buffer);
delay_ms(500);
array[x]=temp;
x=x+1;
break;

case 7:
lcd_gotoxy(x,0);
sprintf(lcd_buffer,"%d",temp);lcd_puts(lcd_buffer);
delay_ms(500);
array[x]=temp;
x=x+1;
break;

case 8:
lcd_gotoxy(x,0);
sprintf(lcd_buffer,"%d",temp);lcd_puts(lcd_buffer);
delay_ms(500);
array[x]=temp;
x=x+1;
break;

case 9:
lcd_gotoxy(x,0);
sprintf(lcd_buffer,"%d",temp);lcd_puts(lcd_buffer);
delay_ms(500);
array[x]=temp;
x=x+1;
break;

case '0':
lcd_gotoxy(x,0);
temp=0;
lcd_putsf("0");
delay_ms(500);
array[x]=temp;
x=x+1;
        break;

case 'A':
goto start;
break;

case 'B':
break;
```

```
case 'C':
break;

case 'D':
break;

case '*':
break;

case '#':
break;
}

if(x==11)
goto next;
goto awal;

next:
//===================== nilai input
lcd_gotoxy(0,1);
input=array[8]*100 + array[9]*10 + array[10];
sprintf(lcd_buffer,"Input = %d  cm",input);
lcd_puts(lcd_buffer);
delay_ms(1000);

//===================== nilai input dikonversi ke satuan hitung putaran roda
//===================== 1 satuan hitung = 1.35 cm
satuan=input/1.35;

//===================== nilai awal
compass();
cmps_0 = cmps;
rotary_awal();

p = 0;
jumlah = 0;

//====================================== start (rotary1)
next1:
ping_0();
ping_1();
ping_2();

if(ping1 > 100)
{
	if(ping0 <= ping2)
	{
		if(ping0 <= 30)
		{
			OCR1A=1125;
			OCR1B=1225;
		}
		else if(ping0 > 30 && ping0 <= 70)
		{
			OCR1A=1075;
			OCR1B=1225;
		}
		else if(ping0 > 70 && ping0 <= 120)
		{
			OCR1A=1025;
			OCR1B=1175;
		}
		else if(ping0 > 120)
		{
			OCR1A=1025;
			OCR1B=1225;
			}
		}
		else if(ping0 > ping2)
		{
			if(ping2 <= 30)
			{
				OCR1A=1025;
				OCR1B=1125;
			}
			else if(ping2 > 30 && ping2 <= 70)
			{
				OCR1A=1025;
				OCR1B=1175;
			}
			else if(ping2 > 70 && ping2 <= 120)
			{
				OCR1A=1075;
				OCR1B=1225;
			}
			else if(ping2 > 120)
			{
				OCR1A=1025;
				OCR1B=1225;
			}
		}
	}

	else if(ping1 <= 100)
	{
		if(ping0 <= ping2)
		{
			hadap_kanan();
			jumlah = jumlah + hitung_rotR[p];
			rotary_awal();
			p = p + 1;
			goto milih_kanan_biasa;
		}
		else if(ping0 > ping2)
		{
			hadap_kiri();
			jumlah = jumlah + hitung_rotR[p];
			rotary_awal();
			p = p + 1;
			goto milih_kiri_biasa;
		}
	}

	rotaryR();
	hitung_rotR[p] = rotR;

	total = jumlah + hitung_rotR[p];
	lcd_clear();
	sprintf(lcd_buffer,"pulsa = %d",total);
	lcd_puts(lcd_buffer);

	if(total >= satuan)
	{
	goto selesai;
	}

	goto next1;


	//====================================================== selesai
	selesai:
	OCR1A=1125;
	OCR1B=1125;
```

```
goto start;


//============================================== ke kanan (rotary2)
milih_kanan_biasa:
ping_0();
ping_1();
ping_2();

if(ping1 > 100)
{
	if(ping2 <= 30)
	{
		OCR1A=1125;
		OCR1B=1125;
	}
	else
	{
		if(ping0 <= 30)
		{
			OCR1A=1125;
			OCR1B=1225;
		}
		else if(ping0 > 30 && ping0 <= 70)
		{
			OCR1A=1075;
			OCR1B=1225;
		}
		else if(ping0 > 70 && ping0 <= 120)
		{
			OCR1A=1025;
			OCR1B=1175;
		}
		else if(ping0 > 120)
		{
			belok_kiri();
			jumlah = jumlah + 6 + 9;
			rotary_awal();
			p = p + 1;
			goto milih_kanan_itung;
		}
	}
}

else if(ping1 <= 100)
{
	OCR1A=1125;
	OCR1B=1125;
}

rotaryR();
hitung_rotR[p] = rotR;

goto milih_kanan_biasa;


//==================================== ke kanan hitung (rotary3)
milih_kanan_itung:
ping_0();
ping_1();
ping_2();

if(ping1 > 100)
{
	if(ping2 <= 30)
	{
		OCR1A=1125;
```

```
        OCR1B=1125;
    }
    else
    {
        if(ping0 <= 30)
        {
            OCR1A=1125;
            OCR1B=1225;
        }
        else if(ping0 > 30 && ping0 <= 70)
        {
            OCR1A=1075;
            OCR1B=1225;
        }
        else if(ping0 > 70 && ping0 <= 120)
        {
            OCR1A=1025;
            OCR1B=1175;
        }
        else if(ping0 > 120)
        {
            belok_kiri();
            jumlah = jumlah + hitung_rotR[p];
            rotary_awal();
            p = p + 1;
            goto milih_kanan_biasa_2;
        }
    }
}

else if(ping1 <= 100)
{
    OCR1A=1125;
    OCR1B=1125;
}

rotaryR();
hitung_rotR[p] = rotR;

total = jumlah + hitung_rotR[p];
lcd_clear();
sprintf(lcd_buffer,"pulsa = %d",total);
lcd_puts(lcd_buffer);
if(total >= satuan)
{
goto selesai;
}

goto milih_kanan_itung;


//==================================== kembali ke jalur (rotary4)
milih_kanan_biasa_2:
ping_0();
ping_1();
ping_2();

if(ping1 > 100)
{
    if(ping2 <= 30)
    {
        OCR1A=1125;
        OCR1B=1125;
    }
    else
    {
        if(ping0 <= 30)
        {
            OCR1A=1125;
            OCR1B=1225;
        }
        else if(ping0 > 30 && ping0 <= 70)
        {
            OCR1A=1075;
            OCR1B=1225;
        }
        else if(ping0 > 70 && ping0 <= 120)
        {
            OCR1A=1025;
            OCR1B=1175;
        }
        else if(ping0 > 120)
        {
            OCR1A=1025;
            OCR1B=1225;
        }
    }
}

else if(ping1 <= 100)
{
    OCR1A=1125;
    OCR1B=1125;
}

rotaryR();
hitung_rotR[p] = rotR + 6;


if(hitung_rotR[p] == hitung_rotR[p-2])
{
    hadap_kanan_special();
    jumlah = jumlah + 9;
    rotary_awal();
    p = 0;
    goto next1;
}

goto milih_kanan_biasa_2;

//====================================


//============================================= ke kiri (rotary2)
milih_kiri_biasa:
ping_0();
ping_1();
ping_2();

if(ping1 > 100)
{
    if(ping0 <= 30)
    {
        OCR1A=1125;
        OCR1B=1125;
    }
    else
    {
        if(ping2 <= 30)
        {
            OCR1A=1025;
            OCR1B=1125;
        }
        else if(ping2 > 30 && ping2 <= 70)
        {
            OCR1A=1025;
            OCR1B=1175;
        }
        else if(ping2 > 70 && ping2 <= 120)
        {
            OCR1A=1075;
            OCR1B=1225;
        }
        else if(ping2 > 120)
        {
            belok_kanan();
            jumlah = jumlah + 6 + 9;
            rotary_awal();
            p = p + 1;
            goto milih_kiri_itung;
        }
    }
}

else if(ping1 <= 100)
{
    OCR1A=1125;
    OCR1B=1125;
}

rotaryR();
hitung_rotR[p] = rotR;

goto milih_kiri_biasa;


//===================================== ke kiri hitung (rotary3)
milih_kiri_itung:
ping_0();
ping_1();
ping_2();

if(ping1 > 100)
{
    if(ping0 <= 30)
    {
        OCR1A=1125;
        OCR1B=1125;
    }
    else
    {
        if(ping2 <= 30)
        {
            OCR1A=1025;
            OCR1B=1125;
        }
        else if(ping2 > 30 && ping2 <= 70)
        {
            OCR1A=1025;
            OCR1B=1175;
        }
        else if(ping2 > 70 && ping2 <= 120)
        {
            OCR1A=1075;
            OCR1B=1225;
        }
        else if(ping2 > 120)
        {
            belok_kanan();
            jumlah = jumlah + hitung_rotR[p];
            rotary_awal();
            p = p + 1;
            goto milih_kiri_biasa_2;
        }
    }
}

else if(ping1 <= 100)
{
    OCR1A=1125;
    OCR1B=1125;
}

rotaryR();
hitung_rotR[p] = rotR;

total = jumlah + hitung_rotR[p];
lcd_clear();
sprintf(lcd_buffer,"pulsa = %d",total);
lcd_puts(lcd_buffer);
if(total >= satuan)
{
goto selesai;
}

goto milih_kiri_itung;


//==================================== kembali ke jalur (rotary4)
milih_kiri_biasa_2:
ping_0();
ping_1();
ping_2();

if(ping1 > 100)
{
    if(ping0 <= 30)
    {
        OCR1A=1125;
        OCR1B=1125;
    }
    else
    {
        if(ping2 <= 30)
        {
            OCR1A=1025;
            OCR1B=1125;
        }
        else if(ping2 > 30 && ping2 <= 70)
        {
            OCR1A=1025;
            OCR1B=1175;
        }
        else if(ping2 > 70 && ping2 <= 120)
        {
            OCR1A=1075;
            OCR1B=1225;
        }
        else if(ping2 > 120)
        {
            OCR1A=1025;
            OCR1B=1225;
        }
    }
}

else if(ping1 <= 100)
{
```

```
                OCR1A=1125;
                OCR1B=1125;
        }

        rotaryR();
        hitung_rotR[p] = rotR + 6;

        if(hitung_rotR[p] == hitung_rotR[p-2])
        {
                hadap_kiri_special();
                jumlah = jumlah + 9;
                rotary_awal();
                p = 0;
                goto next1;
        }

        goto milih_kiri_biasa_2;

        //=====================================

      };
}
```

SUBPROGRAM PENGGUNAAN KEYPAD

```
char scan_keypadB(void)
      {
int scankey;
char keypressed =0;
DDRB = 0x0F;
PORTB = 0xFE;
scankey = PINB&0xf0;
switch (scankey)
   {
    case 0xE0 : keypressed = 1;
    break;
    case 0xD0 : keypressed = 2;
    break;
    case 0xB0 : keypressed = 3;
    break;
    case 0x70 : keypressed = 'A';
    break;
    }
    PORTB = 0xFD;
    delay_ms(10);
scankey = PINB&0xf0;
switch (scankey)
   {
    case 0xE0 : keypressed = 4;
    break;
    case 0xD0 : keypressed = 5;
    break;
    case 0xB0 : keypressed = 6;
    break;
    case 0x70 : keypressed = 'B';
    break;
    }
    PORTB = 0xFB;
    delay_ms(10);
    scankey = PINB&0xf0;
switch (scankey)
   {
    case 0xE0 : keypressed = 7;
    break;
    case 0xD0 : keypressed = 8;
    break;
    case 0xB0 : keypressed = 9;
    break;
    case 0x70 : keypressed = 'C';
    break;
    }
    PORTB = 0xF7;
    delay_ms(10);
    scankey = PINB&0xf0;
switch (scankey)
   {
    case 0xE0 : keypressed = '*';
    break;
    case 0xD0 : keypressed = '0';
    break;
    case 0xB0 : keypressed = '#';
    break;
    case 0x70 : keypressed = 'D';
    break;
    }
       return keypressed;
        }
```

SUBPROGRAM PENGGUNAAN SENSOR PING

```
void ping_0(void)
{
        //========================= ping0 / PING kiri
    us0_mulai:
    us0=0;
    DDRA.0=1;
    PORTA.0=1;
    delay_us(5);
    PORTA.0=0;
    DDRA.0=0;
    PORTA.0=1;

    for(i=0;i<1000;i++)
    {
    if(PINA.0==1)
    goto us0_cek1;
    delay_us(1);
    }
    goto us0_mulai;

    us0_cek1:
    if(PINA.0==0)
    goto us0_cek2;
    us0=us0+1;
    goto us0_cek1;

    us0_cek2:
    ping0=us0*3;
    ping0=ping0/10;
    ping0=ping0*17;
    ping0=ping0/5;
    ping0=ping0/5;
    delay_ms(10);
        return;
}

void ping_1(void)
{
        //========================= ping1 / PING depan
    us1_mulai:
    us1=0;
    DDRA.1=1;
    PORTA.1=1;
    delay_us(5);
    PORTA.1=0;
    DDRA.1=0;
    PORTA.1=1;

    for(i=0;i<1000;i++)
    {
    if(PINA.1==1)
    goto us1_cek1;
    delay_us(1);
    }
    goto us1_mulai;

    us1_cek1:
    if(PINA.1==0)
    goto us1_cek2;
    us1=us1+1;
    goto us1_cek1;

    us1_cek2:
    ping1=us1*3;
```

```
        ping1=ping1/10;
        ping1=ping1*17;
        ping1=ping1/5;
        ping1=ping1/5;
        delay_ms(10);
                return;
}


void ping_2(void)
{
                //========================= ping2 / PING kanan
        us2_mulai:
        us2=0;
        DDRA.2=1;
        PORTA.2=1;
        delay_us(5);
        PORTA.2=0;
        DDRA.2=0;
        PORTA.2=1;

        for(i=0;i<1000;i++)
        {
        if(PINA.2==1)
        goto us2_cek1;
        delay_us(1);
        }
        goto us2_mulai;

        us2_cek1:
        if(PINA.2==0)
        goto us2_cek2;
        us2=us2+1;
        goto us2_cek1;

        us2_cek2:
        ping2=us2*3;
        ping2=ping2/10;
        ping2=ping2*17;
        ping2=ping2/5;
        ping2=ping2/5;
        delay_ms(10);
                return;
}

void ping_lcd(void)
{
                lcd_clear();
        lcd_gotoxy(0,0);
        sprintf(lcd_buffer,"%u  ",ping0);
        lcd_puts(lcd_buffer);
        sprintf(lcd_buffer,"%u  ",ping1);
        lcd_puts(lcd_buffer);
        sprintf(lcd_buffer,"%u  ",ping2);
        lcd_puts(lcd_buffer);
                return;
}
```

SUBPROGRAM PENGGUNAAN SENSOR COMPASS

```
int bil1,bil2;

void compass(void)
{
            i2c_start();
      i2c_write(0xc0);
      i2c_write(2);
      i2c_start();
      i2c_write(0xc1);
      bil1=i2c_read(1);
      bil2=i2c_read(0);
      i2c_stop();

      cmps=bil1*256+bil2;

            return;

}

void compass_lcd(void)
{
            lcd_clear();
      lcd_gotoxy(0,0);
      sprintf(lcd_buffer,"compas = %4d",cmps);
      lcd_puts(lcd_buffer);

            return;
}
```

SUBPROGRAM PENGGUNAAN SENSOR ROTARY ENCODER

```
bit a,b;

void rotary_awal(void)
{
        a=PIND.2;
        b=PIND.3;
        rotL=0;
        rotR=0;
        return;
}


void rotaryL(void)
{
        if(a==0)
    {
     if(PIND.2==1)
      {
     a=1;
     rotL=rotL+1;
      }
     else if(PIND.2==0)
      {
      }
    }
    else if(a==1)
    {
     if(PIND.2==1)
      {
      }
     else if(PIND.2==0)
      {
     a=0;
         rotL=rotL+1;
      }
    }

        return;
}


void rotaryL_lcd(void)
{
        lcd_gotoxy(0,1);
    sprintf(lcd_buffer,"L = %d",rotL);
    lcd_puts(lcd_buffer);
        return;
}


void rotaryR(void)
{
        if(b==0)
    {
     if(PIND.3==1)
      {
     b=1;
     rotR=rotR+1;
      }
     else if(PIND.3==0)
      {
      }
    }
    else if(b==1)
```

```
        {
          if(PIND.3==1)
          {
          }
          else if(PIND.3==0)
          {
          b=0;
                  rotR=rotR+1;
          }
        }

        return;
}


void rotaryR_lcd(void)
{
                lcd_gotoxy(0,0);
        sprintf(lcd_buffer,"R = %d",rotR);
        lcd_puts(lcd_buffer);
                return;
}
```

SUBPROGRAM UNTUK MELAKUKAN MANUVER

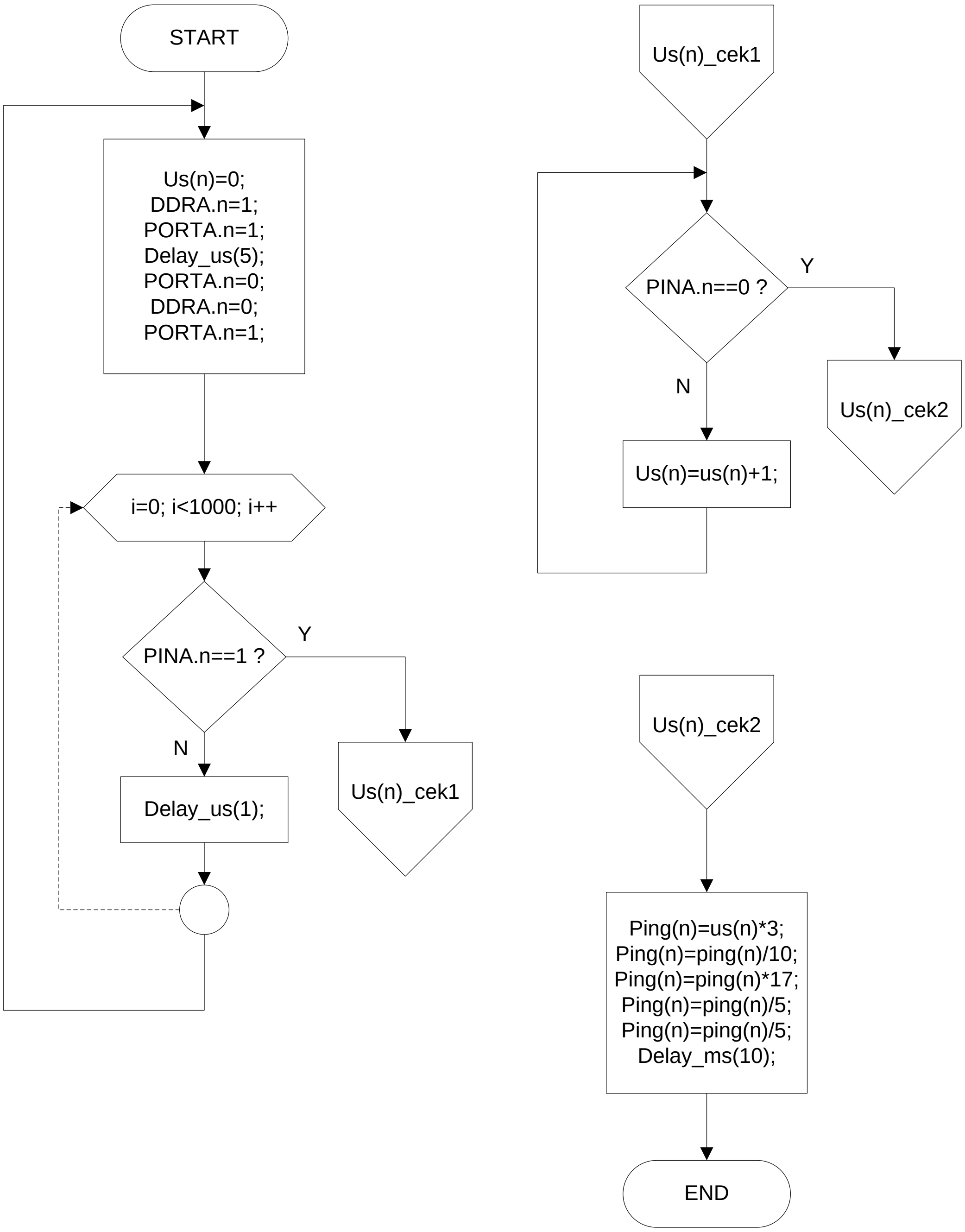
```
void hadap_kiri(void)
{
		PORTA.3 = 1;
		PORTA.4 = 1;

		OCR1A=1025;
		OCR1B=1025;
		rotary_awal();

		ulang_hadap_kiri:
		rotaryR();
		if(rotR >= 9)
		{
		OCR1A=1125;
	OCR1B=1125;
		goto selesai_hadap_kiri;
		}
		goto ulang_hadap_kiri;

		selesai_hadap_kiri:
		PORTA.3 = 0;
		PORTA.4 = 0;
		return;
}


void hadap_kanan(void)
{
		PORTA.3 = 1;
		PORTA.4 = 1;

		OCR1A=1225;
		OCR1B=1225;
		rotary_awal();

		ulang_hadap_kanan:
		rotaryL();
		if(rotL >= 9)
		{
		OCR1A=1125;
	OCR1B=1125;
		goto selesai_hadap_kanan;
		}
		goto ulang_hadap_kanan;

		selesai_hadap_kanan:
		PORTA.3 = 0;
		PORTA.4 = 0;
		return;
}


void belok_kiri(void)
{
	PORTA.3 = 1;
		PORTA.4 = 1;

	compass();
	cmps_1 = cmps;
	rotary_awal();

	OCR1A=1025;
	OCR1B=1140;
```

```
        if(cmps_1 < 900)
                cmps_target = cmps_1 + 2700;
                else if(cmps_1 >= 900)
                cmps_target = cmps_1 - 900;

                ulang_belok_kiri:
                compass();
                cmps_error = cmps - cmps_target;

                if(cmps_error <= 20 && cmps_error >= -20)
                {
                OCR1A=1025;
        OCR1B=1225;
                delay_belok_kiri:
                rotaryR();
                if(rotR >= 6)
                {
                goto selesai_belok_kiri;
                }
                goto delay_belok_kiri;
        }
        goto ulang_belok_kiri;

        selesai_belok_kiri:
        PORTA.3 = 0;
                PORTA.4 = 0;
        return;
}


void belok_kanan(void)
{
        PORTA.3 = 1;
                PORTA.4 = 1;

        compass();
        cmps_1 = cmps;
        rotary_awal();

        OCR1A=1110;
        OCR1B=1225;

        if(cmps_1 < 2700)
                cmps_target = cmps_1 + 900;
                else if(cmps_1 >= 2700)
                cmps_target = cmps_1 - 2700;

                ulang_belok_kanan:
                compass();
                cmps_error = cmps - cmps_target;

                if(cmps_error <= 20 && cmps_error >=-20)
                {
                OCR1A=1025;
        OCR1B=1225;
                delay_belok_kanan:
                rotaryR();
                if(rotR >= 6)
                {
                goto selesai_belok_kanan;
                }
                goto delay_belok_kanan;
        }
        goto ulang_belok_kanan;

        selesai_belok_kanan:
        PORTA.3 = 0;
                PORTA.4 = 0;
        return;
}


void hadap_kiri_special(void)
{
                PORTA.3 = 1;
                PORTA.4 = 1;

                OCR1A=1075;
                OCR1B=1075;

                ulang_hadap_kiri_special:
                compass();
                cmps_error = cmps - cmps_0;

                if(cmps_error <= 20 && cmps_error >= -20)
                {
                OCR1A=1125;
        OCR1B=1125;
                goto selesai_hadap_kiri_special;
                }
                goto ulang_hadap_kiri_special;

                selesai_hadap_kiri_special:
                PORTA.3 = 0;
                PORTA.4 = 0;
                return;
}


void hadap_kanan_special(void)
{
                PORTA.3 = 1;
                PORTA.4 = 1;

                OCR1A=1175;
                OCR1B=1175;

                ulang_hadap_kanan_special:
                compass();
                cmps_error = cmps - cmps_0;

                if(cmps_error <= 20 && cmps_error >=-20)
                {
                OCR1A=1125;
        OCR1B=1125;
                goto selesai_hadap_kanan_special;
                }
                goto ulang_hadap_kanan_special;

                selesai_hadap_kanan_special:
                PORTA.3 = 0;
                PORTA.4 = 0;
                return;
}
```

LAMPIRAN C

DIAGRAM ALIR PROGRAM SENSOR-SENSOR

DIAGRAM ALIR PROGRAM SENSOR PING

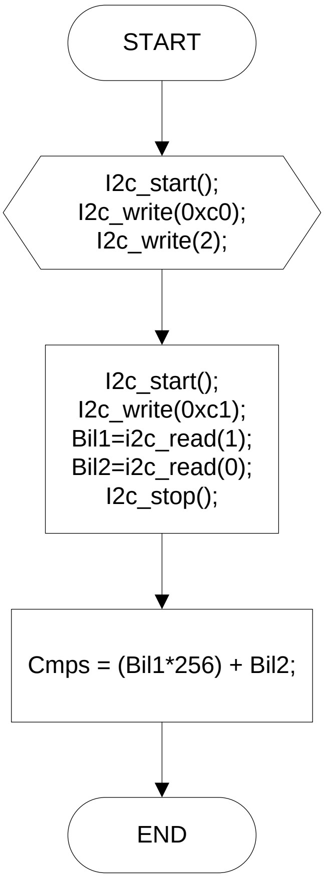


Keterangan :

n =1,2,3 (1 = sensor PING kiri, 2 = sensor PING depan, 3 = sensor PING kanan).

i = variabel pengulangan.

DIAGRAM ALIR PROGRAM SENSOR CMPS03



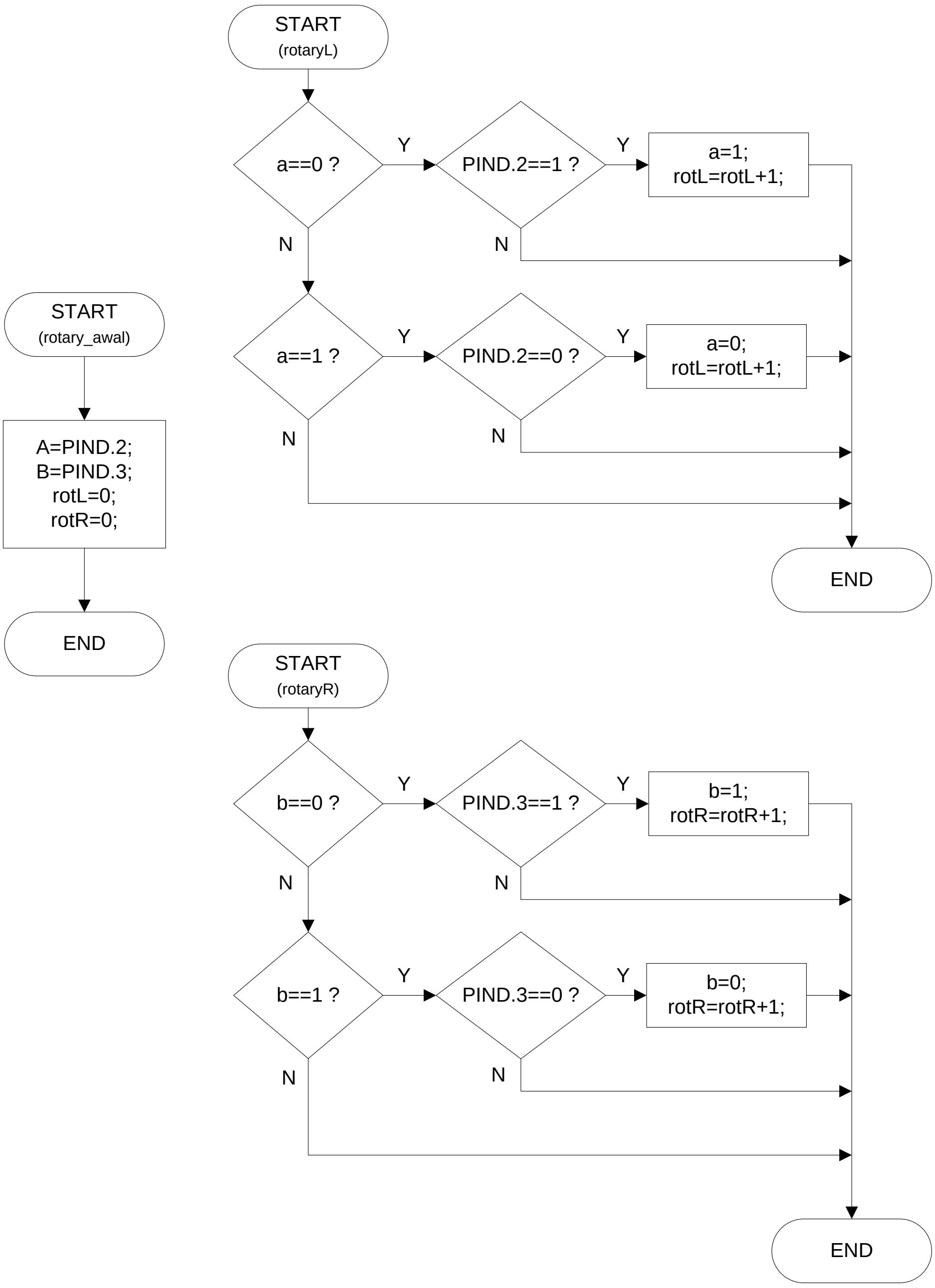
Keterangan :

Bil1 = variabel yang berisi data register 2.

Bil2 = variabel yang berisi data register 3.

Cmps = data sudut arah mata angin dalam 1 *word*.

DIAGRAM ALIR PROGRAM SENSOR ROTARY ENCODER



Keterangan :

Rotary_awal adalah program inisialisasi atau nilai awal sensor *rotary encoder*.

RotaryL adalah program untuk sensor *rotary encoder* pada roda sebelah kiri.

RotaryR adalah program untuk sensor *rotary encoder* pada roda sebelah kanan.

LAMPIRAN D

DATASHEET