


LAMPIRAN 1  
DATA SHEET SERVO GWS S677

## S666/S677/S04

### SERVO SPECIFICATIONS






www.gws.com.tw  
www.gwsus.com

| ITEM         | SIZE<br>mm-in                            | WEIGHT |      | 4.8V    |       |        | 6V      |       |        |
|--------------|--|--------|------|---------|-------|--------|---------|-------|--------|
|              |  |        |      | SPEED   |       | TORQUE | SPEED   |       | TORQUE |
|              |  |        |      | sec/60° | kg/cm | oz/in  | sec/60° | kg/cm | oz/in  |
| S666/FET/STD | 53.0 x 32.0 x 61.6<br>2.14 x 1.26 x 2.43 | 142.4  | 5.02 | 0.25    | 16.00 | 250    | 0.21    | 22.00 | 306    |
| S666/2BB     |  |        |      | 0.22    | 10.40 | 144    | 0.13    | 12.00 | 167    |
| S666N/2BBMG  |  | 180.0  | 6.35 | 0.22    | 18.00 | 250    | 0.18    | 22.00 | 306    |
| S666NF/2BB   |  | 160.4  | 5.66 | 0.19    | 18.50 | 257    | 0.16    | 22.50 | 312    |
| S666NXF/2BB  |  |        |      | 0.15    | 15.00 | 208    | 0.13    | 18.00 | 250    |
| S677-2BB/IMG |  | 180.0  | 6.35 | 0.17    | 21.50 | 298    | 0.145   | 26.50 | 368    |
| S049B        | 54.4 x 26.5 x 51.5<br>2.14 x 1.04 x 2.03 | 110.0  | 3.88 | 0.25    | 10.00 | 139    | 0.20    | 13.00 | 181    |

- GWS servos are designed compatible with RC systems manufactured by Futaba, JR, Hitec, Airtronics (Sanwa), Multiplex and any systems using 1500  $\mu$ S neutral.
- When installing the servo with the provided hardware, insert the screws through grommets (and eyelets) and tighten the screws until the grommets are slightly crushed for best shock absorption. Do not over-tighten the screws.
- All GWS servos are warranted against defects in material and Workmanship. This warranty does not apply to any products which have been improperly installed, handled, abused, damaged in crash, nor been repaired or altered by unauthorized agencies. If any service is required, consult with the model shop where you originally purchased the products.
- Specifications are subject to change without notice.
- Please support your LOCAL HOBBY SHOP for your demands, convenience, newest tip, more technical, information etc.

To see the latest technical information about the product, please visit our web site of GWS: [www.gws.com.tw](http://www.gws.com.tw)

Contact: for after service please contact  
 In USA: [service@gwsus.com](mailto:service@gwsus.com)  
 In China: [china@gws.com.tw](mailto:china@gws.com.tw)  
 Others: [service@gws.com.tw](mailto:service@gws.com.tw)

Manufactured by Grand Wing System China, Inc.  
V7.1 PYMS-1098-2-

## LAMPIRAN 2

### DATA SHEET ATMEGA 8535

#### Features

- High-performance, Low-power AVR<sup>®</sup> 8-bit Microcontroller
- Advanced RISC Architecture
  - 130 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 8K Bytes of In-System Self-Programmable Flash  
Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits  
In-System Programming by On-chip Boot Program  
True Read-While-Write Operation
  - 512 Bytes EEPROM  
Endurance: 100,000 Write/Erase Cycles
  - 512 Bytes Internal SRAM
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    - 8 Single-ended Channels
    - 7 Differential Channels for TQFP Package Only
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x for TQFP Package Only
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad QFN/MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega8535L
  - 4.5 - 5.5V for ATmega8535
- Speed Grades
  - 0 - 8 MHz for ATmega8535L
  - 0 - 16 MHz for ATmega8535



**8-bit AVR<sup>®</sup>**  
**Microcontroller**  
**with 8K Bytes**  
**In-System**  
**Programmable**  
**Flash**

**ATmega8535**  
**ATmega8535L**

2552C-AVR-10/98



UNIVERSITAS KATOLIK WIDYA MANDALA  
SURABAYA

## LAMPIRAN 3

### DATA SHEET CMPS03

#### CMPS03 - Robot Compass Module

This compass module has been specifically designed for use in robots as an aid to navigation. The aim was to produce a unique number to represent the direction the robot is facing. The compass uses the Philips KMZ51 magnetic field sensor, which is sensitive enough to detect the Earth's magnetic field. The output from two of them mounted at right angles to each other is used to compute the direction of the horizontal component of the Earth's magnetic field. We have [examples](#) of using the Compass module with a wide range of popular controllers.

Connections to the compass module



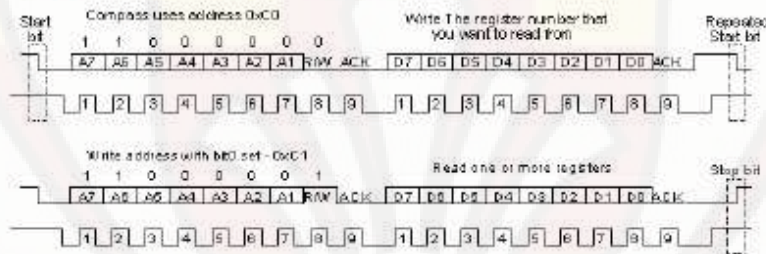
- Pin 9 - 0v Ground
- Pin 8 - No Connect
- Pin 7 - 5000 Hz
- Pin 6 - Calibrate
- Pin 5 - No Connect
- Pin 4 - PWM
- Pin 3 - SDA
- Pin 2 - SCL
- Pin 1 - +5v

The compass module requires a 5v power supply at a nominal 15mA. There are two ways of getting the bearing from the module. A PWM signal is available on pin 4, or an I2C interface is provided on pins 2,3.

The PWM signal is a pulse width modulated signal with the positive width of the pulse representing the angle. The pulse width varies from 1mS ( $0^\circ$ ) to 36.99mS ( $359.9^\circ$ ) - in other words 100uS/ $^\circ$  with a +1mS offset. The signal goes low for 65mS between pulses, so the cycle time is 65mS + the pulse width - ie. 66ms-102ms. The pulse is generated by a 16 bit timer in the processor giving a 1uS resolution, however I would not recommend measuring this to anything better than  $0.1^\circ$  (10uS). Make sure you connect the I2C pins, SCL and SDA, to the 5v supply if you are using the PWM, as there are no pull-up resistors on these pins.



Pin 2,3 are an I2C interface and can be used to get a direct readout of the bearing. If the I2C interface is not used then these pins should be pulled high (to +5v) via a couple of resistors. Around 47k is ok, the values are not at all critical.



I2C communication protocol with the compass module is the same as popular eeprom's such as the 24C04. First send a start bit, the module address (0XC0) with the read/write bit low, then the register number you wish to read. This is followed by a repeated start and the module address again with the read/write bit high (0XC1). You now read one or two bytes for 8bit or 16bit registers respectively. 16bit registers are read high byte first. The compass has a 16 byte array of registers, some of which double up as 16 bit registers as follows;

| Register | Function  |
|----------|---|
| 0        | Software Revision Number  |
| 1        | Compass Bearing as a byte, i.e. 0-255 for a full circle                                 |
| 2,3      | Compass Bearing as a word, i.e. 0-3599 for a full circle, representing 0-359.9 degrees. |
| 4,5      | Internal Test - Sensor1 difference signal - 16 bit signed word                          |
| 6,7      | Internal Test - Sensor2 difference signal - 16 bit signed word                          |
| 8,9      | Internal Test - Calibration value 1 - 16 bit signed word                                |
| 10,11    | Internal Test - Calibration value 2 - 16 bit signed word                                |
| 12       | Unused - Read as Zero   |
| 13       | Unused - Read as Zero   |
| 14       | Unused - Read as Undefined  |
| 15       | Calibrate Command - Write 255 to perform calibration step. See text.                    |

Register 0 is the Software revision number (8 at the time of writing). Register 1 is the bearing converted to a 0-255 value. This may be easier for some applications than 0-360 which requires two bytes. For those who require better resolution registers 2 and 3 (high byte first) are a 16 bit unsigned integer in the range 0-3599. This represents 0-359.9°. Registers 4 to 11 are internal test registers and 12,13 are unused. Register 14 is undefined. Don't read them if you don't want them - you'll just waste your I2C bandwidth. Register 15 is used to calibrate the compass. [Full calibration information is here.](#)

The I2C interface does not have any pull-up resistors on the board, these should be provided elsewhere, most probably with the bus master. They are required on both the SCL and SDA lines, but only once for the whole bus, not on each module. I suggest a value of 1k8 if you are going to be working up to 400KHz and 1k2 or even 1k if you are going up to 1MHz. The compass is designed to work at up to the standard clock speed (SCL) of 100KHz, however the clock speed can be raised to 1MHz providing the following precaution is taken;

At speeds above around 160KHz the CPU cannot respond fast enough to read the I2C data. Therefore a small delay of 50uS should be inserted either side of writing the register address. No delays are required anywhere else in the sequence. By doing this, I have tested the compass module up to 1.3MHz SCL clock speed. There is an example driver [here](#) using the HITECH PICC compiler for the PIC16F877. Note that the above is of no concern if you are using popular embedded language processors such as the [OOPic](#). The compass module always operates as a slave, its never a bus master.

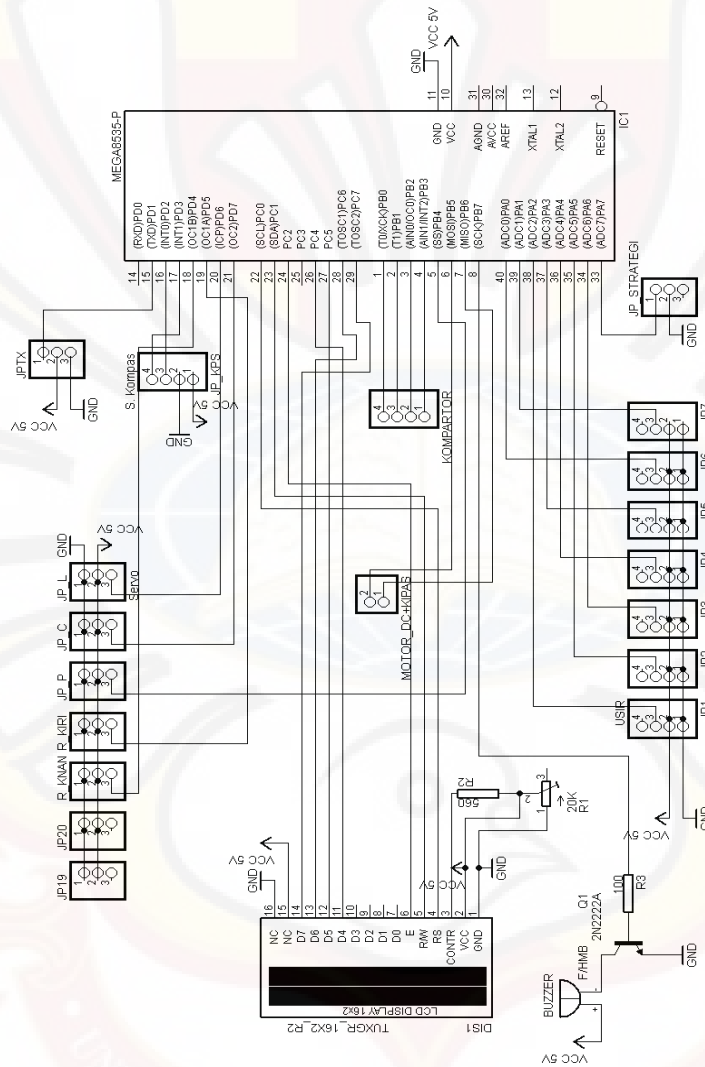
Pin 7 is an input pin selecting either 50Hz (low) or 60Hz (high) operation. I added this option after noticing a jitter of around 1.5° in the output. The cause was the 50Hz mains field in my workshop. By converting in synchronism with the mains frequency this was reduced to around 0.2°. An internal conversion is done every 40mS (50Hz) or every 33.3mS (60Hz). The pin has an on-board pull-up can be left unconnected for 60Hz operation. There is no synchronism between the PWM or I2C outputs and the conversion. They both retrieve the most recent internal reading, which is continuously converted, whether it is used or not.

Pin 6 is used to calibrate the compass. The calibrate input (pin 6) has an on-board pull-up resistor and can be left unconnected after calibration. Calibration is identical to the CMPS01 Rev7 procedure. [Full calibration information is here](#).

Pins 5 and 8 are No Connect. Actually pin 8 is the processor reset line and has an on-board pull-up resistor. It is there so that we can program the processor chip after placement on the PCB.

## LAMPIRAN 4

### GAMBAR RANGKAIAN LENGKAP



**Jalur data I/O Mikrokontroler ATmega 8535**

Arus total rangkaian 0.25A

Arus total 4 Servo GWS 6867\_2BBMG 1.22A

Arus total keseluruhan robot 1.47/1.5A

## LAMPIRAN 5

### LISTING PROGRAM

```

*****/
#include <mega8535.h>
#include <delay.h>

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

// I2C Bus functions
#asm
.equ __i2c_port=0x12 ;PORTD
.equ __sda_bit=2
.equ __scl_bit=3
#endasm
#include <i2c.h>

// Declare your global variables here
#define centerKiri 1500
#define centerKanan 1500

#define servoStop 0
#define servoMaju 1
#define servoMundur 2

#define servoKiriDDR DDRD.4
#define servoKiri PORTD.4
#define servoKananDDR DDRD.5
#define servoKanan PORTD.5

#define servoSensorDDR DDRB.4
#define servoSensor PORTB.4
#define servoLenganDDR DDRD.6
#define servoLengan PORTD.6

```

```

#define servoCapitDDR DDRD.7
#define servoCapit PORTD.7

#define uvTronDDR DDRC.3
#define uvTronPORT PORTC.3
#define uvTronPIN PINC.3

#define usound_pulse_ddr_4 DDRA.4
#define usound_pulse_out_4 PORTA.4
#define usound_pulse_in_4 PINA.4

#define usound_pulse_ddr_2 DDRA.5
#define usound_pulse_out_2 PORTA.5
#define usound_pulse_in_2 PINA.5

#define usound_pulse_ddr_3 DDRA.6
#define usound_pulse_out_3 PORTA.6
#define usound_pulse_in_3 PINA.6

#define usound_pulse_ddr_1 DDRA.2
#define usound_pulse_out_1 PORTA.2
#define usound_pulse_in_1 PINA.2

#define usound_pulse_ddr_7 DDRA.1
#define usound_pulse_out_7 PORTA.1
#define usound_pulse_in_7 PINA.1

#define usound_pulse_ddr_6 DDRA.0
#define usound_pulse_out_6 PORTA.0
#define usound_pulse_in_6 PINA.0

#define usound_pulse_ddr_5 DDRA.3
#define usound_pulse_out_5 PORTA.3
#define usound_pulse_in_5 PINA.3

#define kipas PORTB.5
#define kipasDDR DDRB.5

#define bendera PORTB.6
#define benderaDDR DDRB.6

```



```

#define photo1      PINB.2
#define photo1PORT  PORTB.2
#define photo1DDR   DDRB.2

#define photo2      PINB.3
#define photo2PORT  PORTB.3
#define photo2DDR   DDRB.3

#define line1       PINB.0
#define line1PORT   PORTB.0
#define line1DDR    DDRB.0

#define line2       PINB.1
#define line2PORT   PORTB.1
#define line2DDR    DDRB.1

#define buzzerDDR   DDRB.7
#define buzzer      PORTB.7

#define pwmKecilKi  10
#define pwmKecilKa  10

#define micPin      PINA.7
#define minPORT     PORTA.7
#define micDDR      DDRA.7

#define tol2  150

unsigned int periodeH1,periodeH2,periode;
unsigned int usDepan1,usDepan2,usKanan1,usKanan2,usKiri1,usKiri2,usBlk;
unsigned int sudut;
unsigned char VSensi[8];
unsigned char VAmb;

int kor,atas,bawah,kiri,kanan;
//unsigned int konfRuang,posisiStart;
//unsigned char adaApi,adaBayi;
//unsigned char lilinLantai2 = 255;
//unsigned char posisiBayi = 255;

```

```

// Timer 1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    // Place your code here
    TCNT1H=periode/256;
    TCNT1L=periode%256;

    PORTD.4 = 1;
    PORTD.5 = 1;

}

// Timer 1 output compare A interrupt service routine
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{
    // Place your code here
    PORTD.4 = 0;
}

// Timer 1 output compare B interrupt service routine
interrupt [TIM1_COMPB] void timer1_compb_isr(void)
{
    // Place your code here
    PORTD.5 = 0;
}

void motorKiri(unsigned char arah,unsigned int speed) {
    unsigned int kecepatan;

    if (speed>100) speed=100;
    if (arah==servoMaju) {
        kecepatan = centerKiri + (speed * 9);
        periodeH1 = periode + kecepatan/2;
    }
    else if (arah==servoMundur) {
        kecepatan = centerKiri - (speed * 9);
        periodeH1 = periode + kecepatan/2;
    }
    else {

```

```

        periodeH1 = periode + centerKiri/2;
    }
    OCR1A = periodeH1;
}

void motorKanan(unsigned char arah,unsigned int speed) {
    unsigned int kecepatan;

    if (speed>100) speed=100;
    if (arah==servoMaju) {
        kecepatan = centerKanan - (speed * 9);
        periodeH2 = periode + kecepatan/2;
    }
    else if (arah==servoMundur) {
        kecepatan = centerKanan + (speed * 9);
        periodeH2 = periode + kecepatan/2;
    }
    else {
        periodeH2 = periode + centerKanan/2;
    }
    OCR1B = periodeH2;
}

void initMotor(void) {
    periode = 65535 - 15000/2;
    periodeH1 = periode + centerKiri/2;
    periodeH2 = periode + centerKanan/2;

    // Timer/Counter 1 initialization
    // Clock source: System Clock
    // Clock value: 500.000 kHz
    // Mode: Normal top=FFFFh
    // OC1A output: Discon.
    // OC1B output: Discon.
    // Noise Canceler: Off
    // Input Capture on Falling Edge
    // Timer 1 Overflow Interrupt: On
    // Input Capture Interrupt: Off
    // Compare A Match Interrupt: On
    // Compare B Match Interrupt: On
    TCCR1A=0x00;

```

```

TCCR1B=0x02;
TCNT1 = periode;

OCR1A = periodeH1;
OCR1B = periodeH2;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x1C;
}

void initTimer(void) {
// 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=0x00;
TCCR1B=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
}

void initIO(void) {

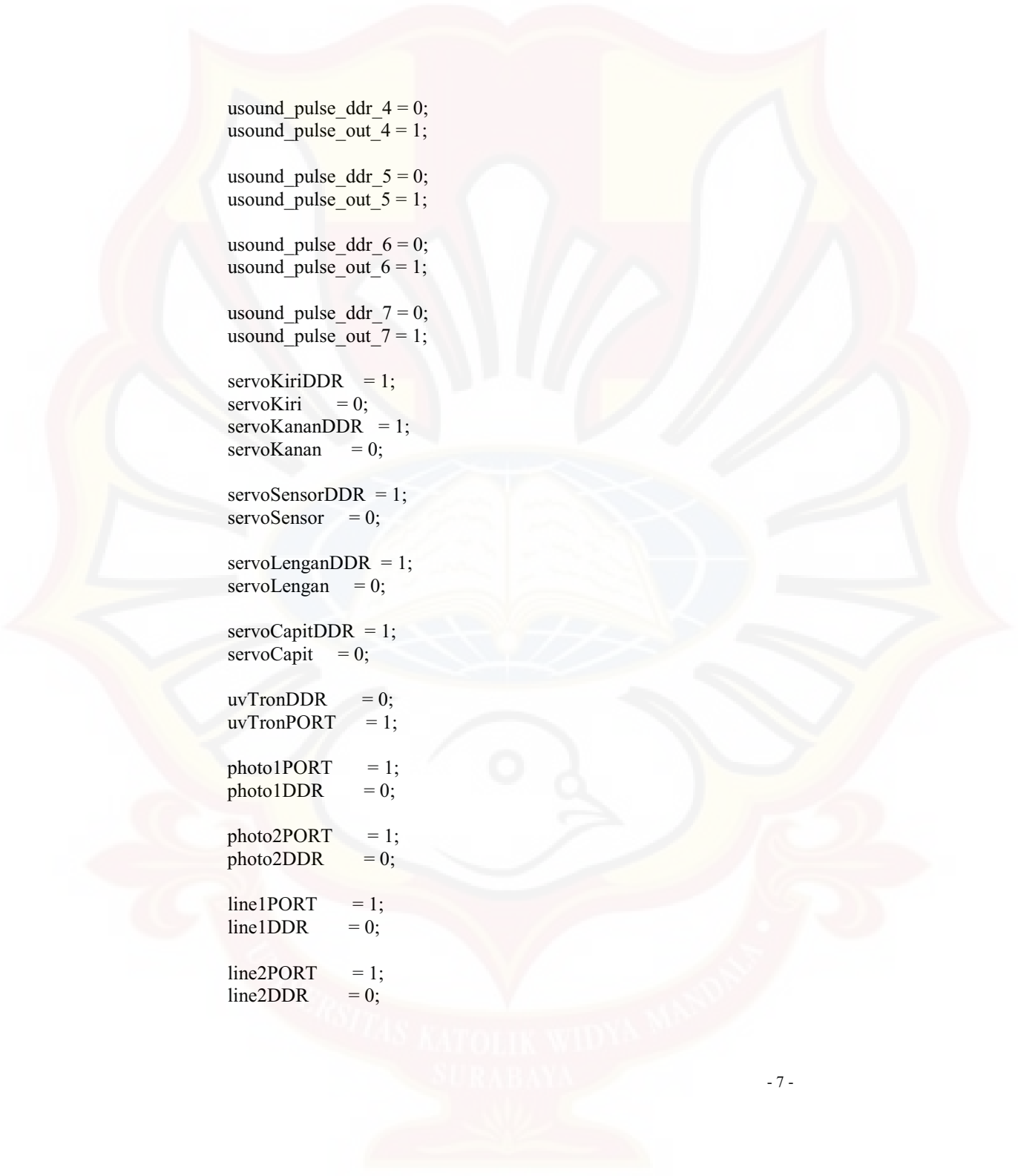
usound_pulse_ddr_1 = 0;
usound_pulse_out_1 = 1;

usound_pulse_ddr_2 = 0;
usound_pulse_out_2 = 1;

usound_pulse_ddr_3 = 0;
usound_pulse_out_3 = 1;

```





```
usound_pulse_ddr_4 = 0;  
usound_pulse_out_4 = 1;
```

```
usound_pulse_ddr_5 = 0;  
usound_pulse_out_5 = 1;
```

```
usound_pulse_ddr_6 = 0;  
usound_pulse_out_6 = 1;
```

```
usound_pulse_ddr_7 = 0;  
usound_pulse_out_7 = 1;
```

```
servoKiriDDR = 1;  
servoKiri = 0;  
servoKananDDR = 1;  
servoKanan = 0;
```

```
servoSensorDDR = 1;  
servoSensor = 0;
```

```
servoLenganDDR = 1;  
servoLengan = 0;
```

```
servoCapitDDR = 1;  
servoCapit = 0;
```

```
uvTronDDR = 0;  
uvTronPORT = 1;
```

```
photo1PORT = 1;  
photo1DDR = 0;
```

```
photo2PORT = 1;  
photo2DDR = 0;
```

```
line1PORT = 1;  
line1DDR = 0;
```

```
line2PORT = 1;  
line2DDR = 0;
```

```

        kipas      = 0;
        kipasDDR   = 1;

        bendera    = 0;
        benderaDDR = 1;

        buzzerDDR  = 1;
        buzzer     = 0;

        minPORT = 1;
        micDDR   = 0;
    }

    int bacaKompas(void) {
        int dataKompas;
        i2c_start();          delay_us(10);
        i2c_write(0xC0);       delay_us(10);
        i2c_write(0x02);       delay_us(10);
        i2c_start();          delay_us(10);
        i2c_write(0xC1);       delay_us(10);
        dataKompas=256*i2c_read(1); delay_us(10);
        dataKompas+=i2c_read(0); delay_us(10);
        i2c_stop();
        return dataKompas;
    }

    void belokKeSudut(int sudutTujuan) {
        int sudutskrg,error1,tol,pwm;
        int kp,kd,ki,errorLama,dError,sError,counter;
        char countKoreksi;

        lcd_gotoxy(0,1);
        lcd_putsf("Belok ke sudut ");

        kp = 2;
        kd = 0;
        ki = 0;

        errorLama = 0;
        dError = 0;

```

```

sError = 0;
counter = 0;

countKoreksi = 1;
tol=30;
if (sudutTujuan>3599) sudutTujuan=sudutTujuan-3599;
if (sudutTujuan<0) sudutTujuan=3599+sudutTujuan;

cek_sudut:
delay_ms(10);
sudutskrg=bacaKompas();
error1 = sudutskrg-sudutTujuan;

if (error1>1800)
{
    error1=-((3600-error1));
}
else if (error1 <-1800)
{
    error1=error1+3600;
}

if ((error1>tol) || (error1<-tol)) {

    dError = errorLama - error1 ;

    if (dError<tol) counter++;
    if (counter>1) {sError++;counter=0;}

    errorLama = error1;

    pwm = (kp * (error1/30)) + (kd * dError/30);

    if (pwm>0) {
        pwm = pwm + (ki * sError);
        if (pwm<5) pwm=5;
        if (pwm>60) pwm=50;
        goto putarKeKiri;
    }
    else {

```

```

        pwm = - pwm;
        pwm = pwm + (ki * sError);
        if (pwm<5) pwm=5;
        if (pwm>60) pwm=60;
        goto putarKeKanan;
    }
}
else {
    goto keluar1;
}

putarKeKanan:

    motorKiri(servoMaju,pwm);
    motorKanan(servoMundur,pwm);
    goto cek_sudut;

putarKeKiri:

    motorKiri(servoMundur,pwm);
    motorKanan(servoMaju,pwm);
    goto cek_sudut;

keluar1:
    motorKiri(servoStop,100);
    motorKanan(servoStop,100);

    delay_ms(100);

    sudutskrg=bacaKompas();
    error1=sudutTujuan-sudutskrg;

    if (error1>1800) {
        error1=- (3600-error1);
    }
    else if (error1 <-1800) {
        error1=error1+3600;
    }
}

```



```

    if (countKoreksi) {
        countKoreksi--;
        if ((error1>tol)||((error1<-tol)) {goto cek_sudut;}}
    }

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);

}

void beep(unsigned char kali) {

    unsigned char i;

    for (i=0;i<kali;i++) {
        delay_ms(100);
        buzzer = 1;
        delay_ms(100);
        buzzer = 0;
    }

}

void cekKompas(void) {
    unsigned int sudutStart;
    while (1) {

        sudutStart=bacaKompas();

        lcd_gotoxy(0,1);
        lcd_putsf("sudut = ");
        lcd_putchar(sudutStart/1000%10 + 0x30);
        lcd_putchar(sudutStart/100 %10 + 0x30);
        lcd_putchar(sudutStart/10%10 + 0x30);
        lcd_putchar(sudutStart%10 + 0x30);

        //putchar(13);
        delay_ms(500);
    }
}

```

```

unsigned int  read_usound_depan1()
{
    unsigned int  US,timeout;

    TIFR |= 0x40;

    usound_pulse_ddr_1 = 1;  // set SIG pin as output

    usound_pulse_out_1 = 0;  //
    delay_us(50);            // send start pulse
    usound_pulse_out_1 = 1;  //

    usound_pulse_ddr_1 = 0;  // set SIG pin as input

    TCNT2=0;
    timeout = 30000;
    while ((usound_pulse_in_1) && (timeout>0)) { // wait for return pulse
        timeout--;
    }
    TCCR2=0x06;            // start timer (timer period = 64uS)
    while ((!usound_pulse_in_1) && !(TIFR & 0x40) && (TCNT2<255)
); // timing the return pulse
    TCCR2=0x00;            // stop timer
    US = TCNT2;            // save timer value to variable US

    return US;
/*
    unsigned int  US,timeout;

    TIFR |= 0x40;

    usound_pulse_ddr_1 = 1;  // set SIG pin as output

    usound_pulse_out_1 = 0;  //
    delay_us(50);            // send start pulse
    usound_pulse_out_1 = 1;  //

    usound_pulse_ddr_1 = 0;  // set SIG pin as input

    TCNT2=0;

```

```

        timeout = 30000;
        while ((usound_pulse_in_1) && (timeout>0)) { // wait for return pulse
            timeout--;
        }
        TCCR2=0x06;          // start timer (timer period = 64uS)
        while ((!usound_pulse_in_1) && !(TIFR & 0x40) && (TCNT2<255)
    ); // timing the return pulse
        TCCR2=0x00;          // stop timer
        US = TCNT2;           // save timer value to variable US

        //US = US*10/64;      // convert to millimeters

        return US;
    */
}

unsigned int  read_usound_depan2()
{
    unsigned int  US,timeout;

    TIFR |= 0x40;

    usound_pulse_ddr_2 = 1; // set SIG pin as output

    usound_pulse_out_2 = 0; //
    delay_us(50);           // send start pulse
    usound_pulse_out_2 = 1; //

    usound_pulse_ddr_2 = 0; // set SIG pin as input

    TCNT2=0;
    timeout = 30000;
    while ((usound_pulse_in_2) && (timeout>0)) { // wait for return pulse
        timeout--;
    }
    TCCR2=0x06;          // start timer (timer period = 64uS)
    while ((!usound_pulse_in_2) && !(TIFR & 0x40) && (TCNT2<255)
); // timing the return pulse
    TCCR2=0x00;          // stop timer
    US = TCNT2;           // save timer value to variable US

```

```

        return US;
    }

    unsigned int  read_usound_kanan1()
    {
        unsigned int  US,timeout;

        TIFR |= 0x40;

        usound_pulse_ddr_3 = 1;  // set SIG pin as output

        usound_pulse_out_3 = 0;  //
        delay_us(50);            // send start pulse
        usound_pulse_out_3 = 1;  //

        usound_pulse_ddr_3 = 0;  // set SIG pin as input

        TCNT2=0;
        timeout = 30000;
        while ((usound_pulse_in_3) && (timeout>0)) { // wait for return pulse
            timeout--;
        }
        TCCR2=0x06;           // start timer (timer period = 64uS)
        while ((!usound_pulse_in_3) && !(TIFR & 0x40) && (TCNT2<255)
    ); // timing the return pulse
        TCCR2=0x00;           // stop timer
        US = TCNT2;            // save timer value to variable US

        return US;
    }

    unsigned int  read_usound_kanan2()
    {
        unsigned int  US,timeout;

        TIFR |= 0x40;

        usound_pulse_ddr_4 = 1;  // set SIG pin as output

```



```

usound_pulse_out_4 = 0; //
delay_us(50);           // send start pulse
usound_pulse_out_4 = 1; //

usound_pulse_ddr_4 = 0; // set SIG pin as input

TCNT2=0;
timeout = 30000;
while ((usound_pulse_in_4) && (timeout>0)) { // wait for return pulse
    timeout--;
}
TCCR2=0x06;           // start timer (timer period = 64uS)
while (!(usound_pulse_in_4) && !(TIFR & 0x40) && (TCNT2<255));
// timing the return pulse
TCCR2=0x00;           // stop timer
US = TCNT2;           // save timer value to variable US

return US;
}

unsigned int read_usound_belakang()
{
    unsigned int US,timeout;

    TIFR |= 0x40;

    usound_pulse_ddr_5 = 1; // set SIG pin as output

    usound_pulse_out_5 = 0; //
    delay_us(50);           // send start pulse
    usound_pulse_out_5 = 1; //

    usound_pulse_ddr_5 = 0; // set SIG pin as input

    TCNT2=0;
    timeout = 30000;
    while ((usound_pulse_in_5) && (timeout>0)) { // wait for return pulse
        timeout--;
    }
    TCCR2=0x06;           // start timer (timer period = 64uS)

```

```

        while ((!usound_pulse_in_5) && !(TIFR & 0x40) &&
(TCNT2<255)); // timing the return pulse
        TCCR2=0x00; // stop timer
        US = TCNT2; // save timer value to variable US

        return US;
    }

    unsigned int read_usound_kiri2()
    {
        unsigned int US,timeout;

        TIFR |= 0x40;

        usound_pulse_ddr_6 = 1; // set SIG pin as output

        usound_pulse_out_6 = 0; //
        delay_us(50); // send start pulse
        usound_pulse_out_6 = 1; //

        usound_pulse_ddr_6 = 0; // set SIG pin as input

        TCNT2=0;
        timeout = 30000;
        while ((usound_pulse_in_6) && (timeout>0)) { // wait for return pulse
            timeout--;
        }
        TCCR2=0x06; // start timer (timer period = 64uS)
        while ((!usound_pulse_in_6) && !(TIFR & 0x40) &&
(TCNT2<255)); // timing the return pulse
        TCCR2=0x00; // stop timer
        US = TCNT2; // save timer value to variable US

        return US;
    }

    unsigned int read_usound_kiri1()
    {
        unsigned int US,timeout;

```

```

TIFR |= 0x40;

usound_pulse_ddr_7 = 1; // set SIG pin as output

usound_pulse_out_7 = 0; //
delay_us(50);           // send start pulse
usound_pulse_out_7 = 1; //

usound_pulse_ddr_7 = 0; // set SIG pin as input

TCNT2=0;
timeout = 30000;
while ((usound_pulse_in_7) && (timeout>0)) { // wait for return pulse
    timeout--;
}
TCCR2=0x06;           // start timer (timer period = 64uS)
while ((!usound_pulse_in_7) && !(TIFR & 0x40) &&
(TCNT2<255)); // timing the return pulse
TCCR2=0x00;           // stop timer
US = TCNT2;           // save timer value to variable US

return US;
}

void cekUS(void) {
    unsigned int sudutStart;

    while (1) {

        usDepan1 = read_usound_depan1();
        usDepan2 = read_usound_depan2();
        usKanan1 = read_usound_kanan1();
        usKanan2 = read_usound_kanan2();
        usBlk = read_usound_belakang();
        usKiri2 = read_usound_kiri2();
        usKiri1 = read_usound_kiri1();

        sudutStart=bacaKompas();
    }
}

```

```

lcd_gotoxy(0,0);
lcd_putchar(usDepan1/100 %10 + 0x30);
lcd_putchar(usDepan1/10%10 + 0x30);
lcd_putchar(usDepan1%10 + 0x30);

lcd_putchar(' ');
lcd_putchar(usDepan2/100 %10 + 0x30);
lcd_putchar(usDepan2/10%10 + 0x30);
lcd_putchar(usDepan2%10 + 0x30);

lcd_putchar(' ');
lcd_putchar(usKanan1/100 %10 + 0x30);
lcd_putchar(usKanan1/10%10 + 0x30);
lcd_putchar(usKanan1%10 + 0x30);

lcd_putchar(' ');
lcd_putchar(usKanan2/100 %10 + 0x30);
lcd_putchar(usKanan2/10%10 + 0x30);
lcd_putchar(usKanan2%10 + 0x30);

lcd_gotoxy(0,1);
lcd_putchar(usBlk/100 %10 + 0x30);
lcd_putchar(usBlk/10%10 + 0x30);
lcd_putchar(usBlk%10 + 0x30);

lcd_putchar(' ');
lcd_putchar(usKiri2/100 %10 + 0x30);
lcd_putchar(usKiri2/10%10 + 0x30);
lcd_putchar(usKiri2%10 + 0x30);

lcd_putchar(' ');
lcd_putchar(usKiri1/100 %10 + 0x30);
lcd_putchar(usKiri1/10%10 + 0x30);
lcd_putchar(usKiri1%10 + 0x30);

lcd_putchar(' ');
lcd_putchar(sudutStart/1000 %10 + 0x30);
lcd_putchar(sudutStart/100 %10 + 0x30);
lcd_putchar(sudutStart/10%10 + 0x30);
lcd_putchar(sudutStart%10 + 0x30);

```



```

        //putchar(13);
        delay_ms(50);  }
    }

void gerakSensor(unsigned int sudut, char pulsa)
{
    unsigned int n;
    char i;

    initTimer();

    n = (sudut * 40/9 ) + 200;
    //n= 1150;
    i=0;
    while (i<pulsa) {
        servoSensor = 1;
        //delay_us(1500);
        TIFR |= 0x04;
        TCNT1 = 65535 - n;
        TCCR1B=0x02;
        while (!(TIFR & 0x04)) {}
        TCCR1B=0x00;
        servoSensor = 0;
        TIFR |= 0x04;
        //delay_ms(10);
        TCNT1 = 65535 - 15000/2;
        TCCR1B = 0x02;
        while (!(TIFR & 0x04)) {}
        TCCR1B=0x00;
        TIFR |= 0x04;
        i++;
    }
    initMotor();
}

void wallKananSampaiKiri1Dekat(void) {
    int pwm1,pwm2,tol;
    int error;

    loop_wallKananSampaiKiri1Dekat:
    tol = 20;

```

```

motorKiri(servoMaju,100);
motorKanan(servoMaju,100);

do {
    delay_ms(50);
    usKanan1 = read_usound_kanan1()*10;
    usKiri1 = read_usound_kiri1()*10;

    error = 180 - usKanan1;

    if ((error>tol) || (error<-tol)) {

        if (error>0) {
            //koreksi = error*20;
            pwm1 = 3;//pwmKecilKi ;
            pwm2 = 100;
        }
        else {
            //koreksi = (-error)*20;
            pwm1 = 100;
            pwm2 = 5;
        }

        if (pwm1>100) pwm1=100;
        if (pwm1<0) pwm1=0;
        if (pwm2>100) pwm2=100;
        if (pwm2<0) pwm2=0;

        motorKiri(servoMaju,pwm1);
        motorKanan(servoMaju,pwm2);

    }
    else {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);
    }

} while ((usKiri1>300));

```

```

motorKiri(servoStop,100);
motorKanan(servoStop,100);

initTimer();
delay_ms(200);
usKiri1 = read_usound_kiri1()*10;
initMotor();
if (usKiri1>300) goto loop_wallKananSampaiKiri1Dekat;
}

void wallKananSampaiKanan2Jauh(void) {      // KANAN BELAKANG
int pwm1,pwm2,tol;
int error;

tol = 20;
motorKiri(servoMaju,100);
motorKanan(servoMaju,100);

do {
    delay_ms(50);
    usKanan1 = read_usound_kanan1()*10;
    usKanan2 = read_usound_kanan2()*10;

    error = 180 - usKanan1;

    if ((error>tol) || (error<-tol)) {

        if (error>0) {
            //koreksi = error*20;
            pwm1 = 3;//pwmKecilKi ;
            pwm2 = 100;
        }
        else {
            //koreksi = (-error)*20;
            pwm1 = 100;
            pwm2 = 5;
        }

        if (pwm1>100) pwm1=100;
        if (pwm1<0) pwm1=0;
        if (pwm2>100) pwm2=100;
    }
}

```

```

        if (pwm2<0) pwm2=0;

        motorKiri(servoMaju,pwm1);
        motorKanan(servoMaju,pwm2);
    }
    else {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);
    }
} while ((usKanan2<400));

motorKiri(servoStop,100);
motorKanan(servoStop,100);
}

void wallKananSampaiKanan1Jauh(void) {    // KANAN DEPAN
    int pwm1,pwm2,tol;
    int error;

    lcd_gotoxy(0,1);
    lcd_putsf("WallKa_Ka_1_jauh");

    tol = 20;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(50);
        usKanan1 = read_usound_kanan1()*10;

        error = 180 - usKanan1;

        if ((error>tol) || (error<-tol)) {

            if (error>0) {
                //koreksi = error*20;
                pwm1 = 3;//pwmKecilKi ;
                pwm2 = 100;
            }
        }
    } while (error > tol || error < -tol);
}

```

```

    }
    else {
        //koreksi = (-error)*20;
        pwm1 = 100;
        pwm2 = 5;
    }

    if (pwm1>100) pwm1=100;
    if (pwm1<0) pwm1=0;
    if (pwm2>100) pwm2=100;
    if (pwm2<0) pwm2=0;

    motorKiri(servoMaju,pwm1);
    motorKanan(servoMaju,pwm2);
}
else {
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);
}

} while ((usKanan1<400));

motorKiri(servoStop,100);
motorKanan(servoStop,100);
}

void wallKananSampaiKiri2Jauh(void) {
    int pwm1,pwm2,tol;
    int error;

    tol = 20;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(50);
        usKanan1 = read_usound_kanan1()*10;
        usKiri2 = read_usound_kiri2()*10;

```



```

error = 180 - usKanan1;

if ((error>tol) || (error<=-tol)) {

    if (error>0) {
        //koreksi = error*20;
        pwm1 = 3;//pwmKecilKi ;
        pwm2 = 100;
    }
    else {
        //koreksi = (-error)*20;
        pwm1 = 100;
        pwm2 = 5;
    }

    if (pwm1>100) pwm1=100;
    if (pwm1<0) pwm1=0;
    if (pwm2>100) pwm2=100;
    if (pwm2<0) pwm2=0;

    motorKiri(servoMaju,pwm1);
    motorKanan(servoMaju,pwm2);

}
else {
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);
}

} while ((usKiri2<400));

motorKiri(servoStop,100);
motorKanan(servoStop,100);

}

void wallKananSampaiKiri12Jauh(void) {
    int pwm1,pwm2,tol;

```

```

int error;

tol = 20;
motorKiri(servoMaju,100);
motorKanan(servoMaju,100);

do {
    delay_ms(50);
    usKanan1 = read_usound_kanan1()*10;
    usKiri1 = read_usound_kiri1()*10;
    usKiri2 = read_usound_kiri2()*10;

    error = 180 - usKanan1;

    if ((error>tol) || (error<=-tol)) {

        if (error>0) {
            //koreksi = error*20;
            pwm1 = 3 ;
            pwm2 = 100;
        }
        else {
            //koreksi = (-error)*20;
            pwm1 = 100;
            pwm2 = 5;
        }

        if (pwm1>100) pwm1=100;
        if (pwm1<0) pwm1=0;
        if (pwm2>100) pwm2=100;
        if (pwm2<0) pwm2=0;

        motorKiri(servoMaju,pwm1);
        motorKanan(servoMaju,pwm2);

    }
    else {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);
    }
}

```

```

    } while ((usKiri1<400 || usKiri2<400));

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);

}

void wallKananSampaiDepanDekat(void) {
    int pwm1,pwm2,tol;
    int error;

loop_wallKananSampaiDepanDekat:
    tol = 20;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

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

    do {
        delay_ms(50);
        usDepan1 = read_usound_depan1()*10;
        usKanan1 = read_usound_kanan1()*10;

        lcd_gotoxy(0,0);
        lcd_putchar(usDepan1/1000 %10 + 0x30);
        lcd_putchar(usDepan1/100 %10 + 0x30);
        lcd_putchar(usDepan1/10%10 + 0x30);
        lcd_putchar(usDepan1%10 + 0x30);

        lcd_gotoxy(5,0);
        lcd_putchar(usKanan1/1000 %10 + 0x30);
        lcd_putchar(usKanan1/100 %10 + 0x30);
        lcd_putchar(usKanan1/10%10 + 0x30);
        lcd_putchar(usKanan1%10 + 0x30);

        //usDepan2 = read_usound_depan2()*10;

```

```

error = 180 - usKanan1;

if ((error>tol) || (error<=-tol)) {

    if (error>0) {
        //koreksi = error*20;
        pwm1 = 3;//pwmKecilKi ;
        pwm2 = 100;
    }
    else {
        //koreksi = (-error)*20;
        pwm1 = 100;
        pwm2 = 5;
    }

    if (pwm1>100) pwm1=100;
    if (pwm1<0) pwm1=0;
    if (pwm2>100) pwm2=100;
    if (pwm2<0) pwm2=0;

    motorKiri(servoMaju,pwm1);
    motorKanan(servoMaju,pwm2);

}
else {
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);
}

} while ((usDepan1>200)); //&& (usDepan2>100));

motorKiri(servoStop,100);
motorKanan(servoStop,100);

initTimer();
delay_ms(200);
usDepan1 = read_usound_depan1()*10;
initMotor();
if ((usDepan1>240)) goto loop_wallKananSampaiDepanDekat;

```

```

    }

    void wallKananTunda(unsigned int tunda) {
        int pwm1,pwm2,tol;
        int error;

        tol = 20;
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);

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

        do {
            delay_ms(50);
            usKanan1 = read_usound_kanan1()*10;

            lcd_gotoxy(0,0);
            lcd_putchar(usDepan1/1000 %10 + 0x30);
            lcd_putchar(usDepan1/100 %10 + 0x30);
            lcd_putchar(usDepan1/10%10 + 0x30);
            lcd_putchar(usDepan1%10 + 0x30);

            lcd_gotoxy(5,0);
            lcd_putchar(usKanan1/1000 %10 + 0x30);
            lcd_putchar(usKanan1/100 %10 + 0x30);
            lcd_putchar(usKanan1/10%10 + 0x30);
            lcd_putchar(usKanan1%10 + 0x30);

            //usDepan2 = read_usound_depan2()*10;

            error = 180 - usKanan1;

            if ((error>tol) || (error<=-tol)) {

                if (error>0) {
                    //koreksi = error*20;
                    pwm1 = 3;//pwmKecilKi ;
                    pwm2 = 100;
                }
            }
        }
    }

```

```

else {
    //koreksi = (-error)*20;
    pwm1 = 100;
    pwm2 = 5;
}

if (pwm1>100) pwm1=100;
if (pwm1<0) pwm1=0;
if (pwm2>100) pwm2=100;
if (pwm2<0) pwm2=0;

motorKiri(servoMaju,pwm1);
motorKanan(servoMaju,pwm2);

}
else {
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);
}

tunda--;
} while (tunda>1); //&& (usDepan2>100));

motorKiri(servoStop,100);
motorKanan(servoStop,100);
}

void wallKananGaris() {
    int pwm1,pwm2,tol;
    int error;

    tol = 20;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

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

    do {

```



```

delay_ms(50);
usKanan1 = read_usound_kanan1()*10;

lcd_gotoxy(0,0);
lcd_putchar(usDepan1/1000 %10 + 0x30);
lcd_putchar(usDepan1/100 %10 + 0x30);
lcd_putchar(usDepan1/10%10 + 0x30);
lcd_putchar(usDepan1%10 + 0x30);

lcd_gotoxy(5,0);
lcd_putchar(usKanan1/1000 %10 + 0x30);
lcd_putchar(usKanan1/100 %10 + 0x30);
lcd_putchar(usKanan1/10%10 + 0x30);
lcd_putchar(usKanan1%10 + 0x30);

//usDepan2 = read_usound_depan2()*10;

error = 180 - usKanan1;

if ((error>tol) || (error<=-tol)) {

    if (error>0) {
        //koreksi = error*20;
        pwm1 = 3;//pwmKecilKi ;
        pwm2 = 100;
    }
    else {
        //koreksi = (-error)*20;
        pwm1 = 100;
        pwm2 = 5;
    }

    if (pwm1>100) pwm1=100;
    if (pwm1<0) pwm1=0;
    if (pwm2>100) pwm2=100;
    if (pwm2<0) pwm2=0;

    motorKiri(servoMaju,pwm1);
    motorKanan(servoMaju,pwm2);

}

```

```

        else {
            motorKiri(servoMaju,100);
            motorKanan(servoMaju,100);
        }

    } while ((line1==0) && (line1==0)); //&& (usDepan2>100));

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);
}

void wallKananSampaiKiri12Dekat(void) {
    int pwm1,pwm2,tol;
    int error;

    loop_wallKananSampaiKiri12Dekat:
        tol = 20;
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);

        do {
            delay_ms(50);
            usKanan1 = read_usound_kanan1()*10;
            usKiri1 = read_usound_kiri1()*10;
            usKiri2 = read_usound_kiri1()*10;

            error = 180 - usKanan1;

            if ((error>tol) || (error<-tol)) {

                if (error>0) {
                    //koreksi = error*20;
                    pwm1 = 3;//pwmKecilKi ;
                    pwm2 = 100;
                }
                else {
                    //koreksi = (-error)*20;

```

```

        pwm1 = 100;
        pwm2 = 5;
    }

    if (pwm1>100) pwm1=100;
    if (pwm1<0) pwm1=0;
    if (pwm2>100) pwm2=100;
    if (pwm2<0) pwm2=0;

    motorKiri(servoMaju,pwm1);
    motorKanan(servoMaju,pwm2);

}
else {
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);
}

} while ((usKiri1>300) || (usKiri2>300) );

motorKiri(servoStop,100);
motorKanan(servoStop,100);

initTimer();
delay_ms(200);
usKiri1 = read_usound_kiri1()*10;
usKiri2 = read_usound_kiri1()*10;
initMotor();
if      ((usKiri1>300)      ||      (usKiri2>300))      goto
loop_wallKananSampaiKiri12Dekat;
}

void wallKananSampaiDepanDekatLilin(void) {
    int pwm1,pwm2,tol;
    int error;

```

```

loop_wallKananSampaiDepanDekatLilin:
    tol = 40;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

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

    do {
        delay_ms(50);
        usKanan1 = read_usound_kanan1()*10;

        usDepan1 = read_usound_depan2()*10;

        lcd_gotoxy(0,0);
        lcd_putchar(usDepan1/1000 %10 + 0x30);
        lcd_putchar(usDepan1/100 %10 + 0x30);
        lcd_putchar(usDepan1/10%10 + 0x30);
        lcd_putchar(usDepan1%10 + 0x30);

        lcd_gotoxy(5,0);
        lcd_putchar(usKanan1/1000 %10 + 0x30);
        lcd_putchar(usKanan1/100 %10 + 0x30);
        lcd_putchar(usKanan1/10%10 + 0x30);
        lcd_putchar(usKanan1%10 + 0x30);

        //usDepan2 = read_usound_depan2()*10;

        error = 200 - usKanan1;

        if ((error>tol) || (error<-tol)) {

            if (error>0) {
                //koreksi = error*20;
                pwm1 = 3;//pwmKecilKi ;
                pwm2 = 100;
            }
            else {
                //koreksi = (-error)*20;

```

```

        pwm1 = 100;
        pwm2 = 5;
    }

    if (pwm1>100) pwm1=100;
    if (pwm1<0) pwm1=0;
    if (pwm2>100) pwm2=100;
    if (pwm2<0) pwm2=0;

    motorKiri(servoMaju,pwm1);
    motorKanan(servoMaju,pwm2);

}
else {
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);
}

} while ((usDepan1>310)); //&& (usDepan2>100));

motorKiri(servoStop,100);
motorKanan(servoStop,100);

initTimer();
delay_ms(200);
usDepan1 = read_usound_depan2()*10;
initMotor();
if ((usDepan1>310)) goto loop_wallKananSampaiDepanDekatLilin;
}

void wallKananSampaiDepanDekatKursi(void) {
    int pwm1,pwm2,tol;
    int error;

loop_wallKananSampaiDepanDekatKursi:
    tol = 20;

```

```

motorKiri(servoMaju,100);
motorKanan(servoMaju,100);

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

do {
    delay_ms(50);
    usKanan1 = read_usound_kanan1()*10;

    usDepan1 = read_usound_depan2()*10;

    lcd_gotoxy(0,0);
    lcd_putchar(usDepan1/1000 %10 + 0x30);
    lcd_putchar(usDepan1/100 %10 + 0x30);
    lcd_putchar(usDepan1/10%10 + 0x30);
    lcd_putchar(usDepan1%10 + 0x30);

    lcd_gotoxy(5,0);
    lcd_putchar(usKanan1/1000 %10 + 0x30);
    lcd_putchar(usKanan1/100 %10 + 0x30);
    lcd_putchar(usKanan1/10%10 + 0x30);
    lcd_putchar(usKanan1%10 + 0x30);

    //usDepan2 = read_usound_depan2()*10;

    error = 160 - usKanan1;

    if ((error>tol) || (error<=-tol)) {

        if (error>0) {
            //koreksi = error*20;
            pwm1 = 3;//pwmKecilKi ;
            pwm2 = 100;
        }
        else {
            //koreksi = (-error)*20;
            pwm1 = 100;
            pwm2 = 5;
        }
    }
}

```



```

        if (pwm1>100) pwm1=100;
        if (pwm1<0) pwm1=0;
        if (pwm2>100) pwm2=100;
        if (pwm2<0) pwm2=0;

        motorKiri(servoMaju,pwm1);
        motorKanan(servoMaju,pwm2);

    }
    else {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);
    }

} while ((usDepan1>400)); //&& (usDepan2>100));

motorKiri(servoStop,100);
motorKanan(servoStop,100);

initTimer();
delay_ms(200);
usDepan1 = read_usound_depan2()*10;
initMotor();
if ((usDepan1>400)) goto loop_wallKananSampaiDepanDekatKursi;

}

void wallKiriSampaiKanan2Jauh(void) {
    int pwm1,pwm2,tol;
    int error;

    tol = 20;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

```

```

do {
    delay_ms(50);
    usKiri1 = read_usound_kiri1()*10;
    usKanan2 = read_usound_kanan2()*10;

    error = 180 - usKiri1;

    if ((error>tol) || (error<-tol)) {

        if (error>0) {
            //koreksi = error*20;
            pwm1 = 100;
            pwm2 = 5;
        }
        else {
            //koreksi = (-error)*20;
            pwm1 = 3;
            pwm2 = 100;
        }

        if (pwm1>100) pwm1=100;
        if (pwm1<0) pwm1=0;
        if (pwm2>100) pwm2=100;
        if (pwm2<0) pwm2=0;

        motorKiri(servoMaju,pwm1);
        motorKanan(servoMaju,pwm2);

    }
    else {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);
    }

} while ((usKanan2<400));

motorKiri(servoStop,100);
motorKanan(servoStop,100);

```

```

}

void wallKiriSampaiKiri2Dekat(void) {
    int pwm1,pwm2,tol;
    int error;

loop_wallKiriSampaiKiri2Dekat:
    tol = 20;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(50);
        usKiri1 = read_usound_kiri1()*10;
        usKiri2 = read_usound_kiri2()*10;

        error = 180 - usKiri1;

        if ((error>tol) || (error<=-tol)) {

            if (error>0) {
                //koreksi = error*20;
                pwm1 = 100;
                pwm2 = 5;
            }
            else {
                //koreksi = (-error)*20;
                pwm1 = 3 ;
                pwm2 = 100;
            }

            if (pwm1>100) pwm1=100;
            if (pwm1<0) pwm1=0;
            if (pwm2>100) pwm2=100;
            if (pwm2<0) pwm2=0;

            motorKiri(servoMaju,pwm1);
            motorKanan(servoMaju,pwm2);

        }
        else {

```

```

        motorKanan(servoMaju,100);
        motorKiri(servoMaju,100);

    }

    } while ((usKiri2>300));

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);

    initTimer();
    delay_ms(200);
    usKiri2 = read_usound_kiri2()*10;
    initMotor();
    if ((usKiri2>300)) goto loop_wallKiriSampaiKiri2Dekat;
}

void wallKiriSampaiKiri1Jauh(void) {
    int pwm1,pwm2,tol;
    int error;

    tol = 20;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(50);
        usKiri1 = read_usound_kiri1()*10;

        error = 180 - usKiri1;

        if ((error>tol) || (error<-tol)) {

            if (error>0) {
                //koreksi = error*20;
                pwm1 = 100;
                pwm2 = 5;
            }
            else {

```

```

        //koreksi = (-error)*20;
        pwm1 = 3 ;
        pwm2 = 100;
    }

    if (pwm1>100) pwm1=100;
    if (pwm1<0) pwm1=0;
    if (pwm2>100) pwm2=100;
    if (pwm2<0) pwm2=0;

    motorKiri(servoMaju,pwm1);
    motorKanan(servoMaju,pwm2);

}
else {
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);
}

} while ((usKiri1<400));

motorKiri(servoStop,100);
motorKanan(servoStop,100);
}

void wallKiriSampaiKiri2Jauh(void) {
    int pwm1,pwm2,tol;
    int error;

    tol = 20;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(50);
        usKiri1 = read_usound_kiri1()*10;
        usKiri2 = read_usound_kiri2()*10;

```

```

error = 180 - usKiri1;

if ((error>tol) || (error<=-tol)) {

    if (error>0) {
        //koreksi = error*20;
        pwm1 = 100;
        pwm2 = 5;
    }
    else {
        //koreksi = (-error)*20;
        pwm1 = 3 ;
        pwm2 = 100;
    }

    if (pwm1>100) pwm1=100;
    if (pwm1<0) pwm1=0;
    if (pwm2>100) pwm2=100;
    if (pwm2<0) pwm2=0;

    motorKiri(servoMaju,pwm1);
    motorKanan(servoMaju,pwm2);

}
else {
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);
}

} while ((usKiri2<400));

motorKiri(servoStop,100);
motorKanan(servoStop,100);

}

void wallKiriSampaiKiri12Jauh(void) {
    int pwm1,pwm2,tol;
    int error;

```



```

tol = 20;
motorKiri(servoMaju,100);
motorKanan(servoMaju,100);

do {
    delay_ms(50);
    usKiri1 = read_usound_kiri1()*10;
    usKiri2 = read_usound_kiri2()*10;

    error = 180 - usKiri1;

    if ((error>tol) || (error<=-tol)) {

        if (error>0) {
            //koreksi = error*20;
            pwm1 = 100;
            pwm2 = 5;
        }
        else {
            //koreksi = (-error)*20;
            pwm1 = 3 ;
            pwm2 = 100;
        }

        if (pwm1>100) pwm1=100;
        if (pwm1<0) pwm1=0;
        if (pwm2>100) pwm2=100;
        if (pwm2<0) pwm2=0;

        motorKiri(servoMaju,pwm1);
        motorKanan(servoMaju,pwm2);

    }
    else {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);
    }
}

```

```

    } while (usKiri1<400 || usKiri2<400);

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);

}

void wallKiriSampaiKanan1Dekat(void) {
    int pwm1,pwm2,tol;
    int error;

loop_wallKiriSampaiKanan1Dekat:
    tol = 20;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(50);
        usKiri1 = read_usound_kiri1()*10;
        usKanan1 = read_usound_kanan1()*10;

        error = 180 - usKiri1;

        if ((error>tol) || (error<-tol)) {

            if (error>0) {
                //koreksi = error*20;
                pwm1 = 100;
                pwm2 = 5;
            }
            else {
                //koreksi = (-error)*20;
                pwm1 = 3 ;
                pwm2 = 100;
            }

            if (pwm1>100) pwm1=100;
            if (pwm1<0) pwm1=0;
            if (pwm2>100) pwm2=100;
            if (pwm2<0) pwm2=0;

```

```

        motorKiri(servoMaju,pwm1);
        motorKanan(servoMaju,pwm2);
    }
    else {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);
    }

} while ((usKanan1>300));

motorKiri(servoStop,100);
motorKanan(servoStop,100);

initTimer();
delay_ms(200);
usKanan1 = read_usound_kanan1()*10;
initMotor();
if ((usKanan1>300)) goto loop_wallKiriSampaiKanan1Dekat;
}

void wallKiriSampaiKanan2Dekat(void) {
    int pwm1,pwm2,tol;
    int error;

loop_wallKiriSampaiKanan2Dekat:
    tol = 20;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(50);
        usKiri1 = read_usound_kiri1()*10;
        usKanan2 = read_usound_kanan2()*10;

        error = 180 - usKiri1;

```

```

    if ((error>tol) || (error<=-tol)) {

        if (error>0) {
            //koreksi = error*20;
            pwm1 = 100;
            pwm2 = 5;
        }
        else {
            //koreksi = (-error)*20;
            pwm1 = 3 ;
            pwm2 = 100;
        }

        if (pwm1>100) pwm1=100;
        if (pwm1<0) pwm1=0;
        if (pwm2>100) pwm2=100;
        if (pwm2<0) pwm2=0;

        motorKiri(servoMaju,pwm1);
        motorKanan(servoMaju,pwm2);

    }
    else {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);
    }

} while ((usKanan2>400));

motorKiri(servoStop,100);
motorKanan(servoStop,100);

initTimer();
delay_ms(200);
usKanan2 = read_usound_kanan2()*10;
initMotor();
if ((usKanan2>400)) goto loop_wallKiriSampaiKanan2Dekat;

}

```

```

void wallKiriSampaiKanan12Dekat(void) {
    int pwm1,pwm2,tol;
    int error;

loop_wallKiriSampaiKanan12Dekat:
    tol = 20;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(50);
        usKiri1 = read_usound_kiri1()*10;
        usKanan1 = read_usound_kanan1()*10;
        usKanan2 = read_usound_kanan2()*10;

        error = 180 - usKiri1;

        if ((error>tol) || (error<=-tol)) {

            if (error>0) {
                //koreksi = error*20;
                pwm1 = 100;
                pwm2 = 5;
            }
            else {
                //koreksi = (-error)*20;
                pwm1 = 3 ;
                pwm2 = 100;
            }

            if (pwm1>100) pwm1=100;
            if (pwm1<0) pwm1=0;
            if (pwm2>100) pwm2=100;
            if (pwm2<0) pwm2=0;

            motorKiri(servoMaju,pwm1);
            motorKanan(servoMaju,pwm2);

        }
        else {

```

```

        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);

    }

} while ((usKanan1>300) || (usKanan2>300) );

motorKiri(servoStop,100);
motorKanan(servoStop,100);

initTimer();
delay_ms(200);
usKanan1 = read_usound_kanan1()*10;
usKanan2 = read_usound_kanan2()*10;
initMotor();
if      ((usKanan1>300)      ||      (usKanan2>300))      goto
loop_wallKiriSampaiKanan12Dekat;
}

void wallKiriSampaiDepanDekat(void) {
    int pwm1,pwm2,tol;
    int error;

loop_wallKiriSampaiDepanDekat:
    tol = 20;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {

        delay_ms(25);
        usKiri1 = read_usound_kiri1()*10;
        delay_ms(25);
        usDepan1 = read_usound_depan1()*10;

        //usDepan2 = read_usound_depan2()*10;

        error = 200 - usKiri1;

```

```

if ((error>tol) || (error<=-tol)) {

    if (error>0) {
        //koreksi = error*20;
        pwm1 = 100;
        pwm2 = 5;
    }
    else {
        //koreksi = (-error)*20;
        pwm1 = 3 ;
        pwm2 = 100;
    }

    if (pwm1>100) pwm1=100;
    if (pwm1<0) pwm1=0;
    if (pwm2>100) pwm2=100;
    if (pwm2<0) pwm2=0;

    motorKiri(servoMaju,pwm1);
    motorKanan(servoMaju,pwm2);

}
else {
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);
}

} while ((usDepan1>240));

motorKiri(servoStop,100);
motorKanan(servoStop,100);

initTimer();
delay_ms(200);
usDepan1 = read_usound_depan1()*10;
initMotor();

```



```

        if ((usDepan1>240)) goto loop_wallKiriSampaiDepanDekat;
    }

    void wallKiriTunda(unsigned int tunda) {
        int pwm1,pwm2,tol;
        int error;

        tol = 20;
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);

        do {

            delay_ms(50);
            usKiri1 = read_usound_kiri1()*10;

            //usDepan2 = read_usound_depan2()*10;

            error = 180 - usKiri1;

            if ((error>tol) || (error<-tol)) {

                if (error>0) {
                    //koreksi = error*20;
                    pwm1 = 100;
                    pwm2 = 5;
                }
                else {
                    //koreksi = (-error)*20;
                    pwm1 = 3 ;
                    pwm2 = 100;
                }

                if (pwm1>100) pwm1=100;
                if (pwm1<0) pwm1=0;
                if (pwm2>100) pwm2=100;
                if (pwm2<0) pwm2=0;
            }
        } while (tunda>0);
    }

```

```

        motorKiri(servoMaju,pwm1);
        motorKanan(servoMaju,pwm2);
    }
    else {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);
    }

    tunda--;
    } while ((tunda>1));

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);
}

void wallKiriSampaiDepanDekatLilinKiri(void) {
    int pwm1,pwm2,tol;
    int error;

    tol = 20;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(50);
        usKiri1 = read_usound_kiri1()*10;
        usDepan1 = read_usound_depan1()*10;
        usDepan2 = read_usound_depan2()*10;

        error = 180 - usKiri1;

        if ((error>tol) || (error<-tol)) {

            if (error>0) {
                //koreksi = error*20;
                pwm1 = 100;
                pwm2 = 5;
            }
        }
    } while (error > tol || error < -tol);
}

```

```

    }
    else {
        //koreksi = (-error)*20;
        pwm1 = 3 ;
        pwm2 = 100;
    }

    if (pwm1>100) pwm1=100;
    if (pwm1<0) pwm1=0;
    if (pwm2>100) pwm2=100;
    if (pwm2<0) pwm2=0;

    motorKiri(servoMaju,pwm1);
    motorKanan(servoMaju,pwm2);

}
else {
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);
}

} while (usDepan1>180 || usDepan1>180 );

motorKiri(servoStop,100);
motorKanan(servoStop,100);

}

void wallKiriSampaiDepanDekatLilin(void) {
    int pwm1,pwm2,tol;
    int error;

loop_wallKiriSampaiDepanDekatLilin:
    tol = 20;
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(50);

```

```

usKiri1 = read_usound_kiri1()*10;
usDepan1 = read_usound_depan2()*10;

//usDepan2 = read_usound_depan2()*10;

error = 220 - usKiri1;

if ((error>tol) || (error<=-tol)) {

    if (error>0) {
        //koreksi = error*20;
        pwm1 = 100;
        pwm2 = 5;
    }
    else {
        //koreksi = (-error)*20;
        pwm1 = 3 ;
        pwm2 = 100;
    }

    if (pwm1>100) pwm1=100;
    if (pwm1<0) pwm1=0;
    if (pwm2>100) pwm2=100;
    if (pwm2<0) pwm2=0;

    motorKiri(servoMaju,pwm1);
    motorKanan(servoMaju,pwm2);

}
else {
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

}

} while ((usDepan1>310));

motorKiri(servoStop,100);
motorKanan(servoStop,100);

```

```

initTimer();
delay_ms(200);
usDepan1 = read_usound_depan2()*10;
initMotor();
if ((usDepan1>310)) goto loop_wallKiriSampaiDepanDekatLilin;

}
//===================================================== END OF WALL FOLLOWING
//=====================================================

void majuKompasTunda(unsigned int tunda) {
    int sudutTujuan,sudutSkrng,error,tol;

    lcd_gotoxy(0,1);
    lcd_putsf("mjKompas_ Tunda ");

    tol = 20;
    sudutTujuan = bacaKompas();
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(100);

        sudutSkrng = bacaKompas();
        error = sudutTujuan - sudutSkrng;

        if (error>1800)
        {
            error=- (3600-error);
        }
        else if (error <-1800)
        {
            error=error+3600;
        }

        if (error<=-tol) {
            motorKiri(servoMaju,40);
            motorKanan(servoMaju,100);
        }
        else if (error>tol) {
            motorKiri(servoMaju,100);

```

```

        motorKanan(servoMaju,40);
    }
    else {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);
    }

    tunda--;
} while (tunda>1);

motorKiri(servoStop,100);
motorKanan(servoStop,100);
}

void mundurKompasTunda(unsigned int tunda) {
    int sudutTujuan,sudutSkr,error,tol;

    tol = 20;
    sudutTujuan = bacaKompas();
    motorKiri(servoMundur,100);
    motorKanan(servoMundur,100);

    do {
        delay_ms(100);

        sudutSkr = bacaKompas();
        error = sudutTujuan - sudutSkr;

        if (error>1800)
        {
            error=-(3600-error);
        }
        else if (error <-1800)
        {
            error=error+3600;
        }

        if (error<=-tol) {
            motorKiri(servoMundur,100);
            motorKanan(servoMundur,40);

```

```

    }
    else if (error>tol) {
        motorKiri(servoMundur,40);
        motorKanan(servoMundur,100);
    }
    else {
        motorKiri(servoMundur,100);
        motorKanan(servoMundur,100);
    }

    tunda--;
} while (tunda>1);

motorKiri(servoStop,100);
motorKanan(servoStop,100);

}

void majuKompasSampaiKiri1Dekat(void) {
    int sudutTujuan,sudutSkrng,error,tol;

    tol = 20;
    sudutTujuan = bacaKompas();

loop_majuKompasSampaiKiri1Dekat:

    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(100);
        usKiri1 = read_usound_kiri1()*10;

        sudutSkrng = bacaKompas();
        error = sudutTujuan - sudutSkrng;

        if (error>1800)
        {
            error=- (3600-error);
        }
    }

```

```

else if (error < -1800)
{
    error=error+3600;
}

if (error<=-tol) {
    motorKiri(servoMaju,40);
    motorKanan(servoMaju,100);
}
else if (error>tol) {
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,40);
}
else {
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);
}
} while (usKiri1>400);

motorKiri(servoStop,100);
motorKanan(servoStop,100);

initTimer();
delay_ms(200);
usKiri1 = read_usound_kiri1()*10;
initMotor();
if ((usKiri1>400)) goto loop_majuKompasSampaiKiri1Dekat;
}

void majuKompasSampaiKanan1Dekat(void) {
    int sudutTujuan,sudutSkr,error,tol;

    tol = 20;
    sudutTujuan = bacaKompas();

loop_majuKompasSampaiKanan1Dekat:

    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

```



```

do {
    delay_ms(100);
    usKanan1 = read_usound_kanan1()*10;

    sudutSkrng = bacaKompas();
    error = sudutTujuan - sudutSkrng;

    if (error>1800)
    {
        error--(3600-error);
    }
    else if (error <-1800)
    {
        error=error+3600;
    }

    if (error<=tol) {
        motorKiri(servoMaju,40);
        motorKanan(servoMaju,100);
    }
    else if (error>tol) {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,40);
    }
    else {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);
    }

} while (usKanan1>400);

motorKiri(servoStop,100);
motorKanan(servoStop,100);

initTimer();
delay_ms(200);
usKanan1 = read_usound_kanan1()*10;
initMotor();
if ((usKanan1>400)) goto loop_majuKompasSampaiKanan1Dekat;

}

```

```

void majuKompasSampaiKanan12Dekat(void) {
    int sudutTujuan,sudutSkrng,error,tol;
    tol = 20;
    sudutTujuan = bacaKompas();

    loop_majuKompasSampaiKanan12Dekat:

    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(100);
        usKanan1 = read_usound_kanan1()*10;
        usKanan2 = read_usound_kanan2()*10;

        sudutSkrng = bacaKompas();
        error = sudutTujuan - sudutSkrng;

        if (error>1800)
        {
            error=- (3600-error);
        }
        else if (error <-1800)
        {
            error=error+3600;
        }

        if (error<=tol) {
            motorKiri(servoMaju,40);
            motorKanan(servoMaju,100);
        }
        else if (error>tol) {
            motorKiri(servoMaju,100);
            motorKanan(servoMaju,40);
        }
        else {
            motorKiri(servoMaju,100);
            motorKanan(servoMaju,100);
        }
    }
}

```

```

    } while (usKanan1>400 || usKanan2>400);

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);

    initTimer();
    delay_ms(200);
    usKanan1 = read_usound_kanan1()*10;
    usKanan2 = read_usound_kanan2()*10;
    initMotor();
    if ((usKanan1>400 || usKanan2>400)) goto
loop_majuKompasSampaiKanan12Dekat;
}

void majuKompasSampaiDepanDekat(void) {
    int sudutTujuan,sudutSkrng,error,tol;

    lcd_gotoxy(0,1);
    lcd_putsf("mjKompas_Dpn_Dkt");

    loop_majuKompasSampaiDepanDekat:
    tol = 20;
    sudutTujuan = bacaKompas();
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        //usDepan1 = read_usound_depan1()*10;
        delay_ms(25);
        usDepan2 = read_usound_depan2()*10;

        sudutSkrng = bacaKompas();
        error = sudutTujuan - sudutSkrng;

        if (error>1800)
        {
            error=- (3600-error);
        }
        else if (error <-1800)
        {

```

```

        error=error+3600;
    }

    if (error<=-tol) {
        motorKiri(servoMaju,10);
        motorKanan(servoMaju,100);
    }
    else if (error>tol) {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,10);
    }
    else {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);
    }

} while (usDepan2>240); //&& usDepan2>100);

motorKiri(servoStop,100);
motorKanan(servoStop,100);

initTimer();
delay_ms(200);
usDepan2 = read_usound_depan2()*10;
initMotor();
if ((usDepan2>240)) goto loop_majuKompasSampaiDepanDekat;
}

void majuKompasSampaiDepanDekatBayi(void) {
    int sudutTujuan,sudutSkr,error,tol;

    tol = 20;
    sudutTujuan = bacaKompas();
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(50);

```

```

        usDepan1 = read_usound_depan1()*10;
        usDepan2 = read_usound_depan2()*10;

        sudutSkrng = bacaKompas();
        error = sudutTujuan - sudutSkrng;

        if (error>1800)
        {
            error=- (3600-error);
        }
        else if (error <-1800)
        {
            error=error+3600;
        }

        if (error<=-tol) {
            motorKiri(servoMaju,40);
            motorKanan(servoMaju,100);
        }
        else if (error>tol) {
            motorKiri(servoMaju,100);
            motorKanan(servoMaju,40);
        }
        else {
            motorKiri(servoMaju,100);
            motorKanan(servoMaju,100);
        }

    } while ((usDepan1>280) && (usDepan2>280));

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);
}

void majuKompasSampaiDepanDekatLilin(void) {
    int sudutTujuan,sudutSkrng,error,tol;

    tol = 20;

```

```

    sudutTujuan = bacaKompas();
    motorKiri(servoMaju,80);
    motorKanan(servoMaju,80);

    do {
        delay_ms(50);
        usDepan1 = read_usound_depan1()*10;
        usDepan2 = read_usound_depan2()*10;

        sudutSkrng = bacaKompas();
        error = sudutTujuan - sudutSkrng;

        if (error>1800)
        {
            error=- (3600-error);
        }
        else if (error < -1800)
        {
            error=error+3600;
        }

        if (error<=tol) {
            motorKiri(servoMaju,30);
            motorKanan(servoMaju,80);
        }
        else if (error>tol) {
            motorKiri(servoMaju,80);
            motorKanan(servoMaju,30);
        }
        else {
            motorKiri(servoMaju,80);
            motorKanan(servoMaju,80);
        }

    } while ((usDepan1>220) && (usDepan2>220));

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);
}

```

```

void majuKompasSampaiDepanDekatLilinKanan(void) {
    int sudutTujuan,sudutSkrng,error,tol;

    tol = 20;
    sudutTujuan = bacaKompas();
    motorKiri(servoMaju,80);
    motorKanan(servoMaju,80);

    do {
        delay_ms(50);
        usDepan1 = read_usound_depan1()*10;
        usDepan2 = read_usound_depan2()*10;

        sudutSkrng = bacaKompas();
        error = sudutTujuan - sudutSkrng;

        if (error>1800)
        {
            error=- (3600-error);
        }
        else if (error <-1800)
        {
            error=error+3600;
        }

        if (error<=tol) {
            motorKiri(servoMaju,30);
            motorKanan(servoMaju,80);
        }
        else if (error>tol) {
            motorKiri(servoMaju,80);
            motorKanan(servoMaju,30);
        }
        else {
            motorKiri(servoMaju,80);
            motorKanan(servoMaju,80);
        }
    }
}

```

```

    } while (usDepan2>300);

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);

}

void majuKompasSampaiDepanDekatLilinKiri(void) {
    int sudutTujuan,sudutSkrng,error,tol;

    tol = 20;
    sudutTujuan = bacaKompas();
    motorKiri(servoMaju,80);
    motorKanan(servoMaju,80);

    do {
        delay_ms(50);
        usDepan1 = read_usound_depan1()*10;
        usDepan2 = read_usound_depan2()*10;

        sudutSkrng = bacaKompas();
        error = sudutTujuan - sudutSkrng;

        if (error>1800)
        {
            error=- (3600-error);
        }
        else if (error < -1800)
        {
            error=error+3600;
        }

        if (error< -tol) {
            motorKiri(servoMaju,30);
            motorKanan(servoMaju,80);
        }
        else if (error>tol) {
            motorKiri(servoMaju,80);
            motorKanan(servoMaju,30);
        }
    }
}

```



```

    }
    else {
        motorKiri(servoMaju,80);
        motorKanan(servoMaju,80);
    }

    } while (usDepan1>300);

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);
}

void mundurKompasSampaiKanan1Jauh(void) {
    int sudutTujuan,sudutSkrng,error,tol;

    tol = 20;
    sudutTujuan = bacaKompas();
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(50);
        usKanan1 = read_usound_kanan1()*10;

        sudutSkrng = bacaKompas();
        error = sudutTujuan - sudutSkrng;

        if (error>1800)
        {
            error=- (3600-error);
        }
        else if (error < -1800)
        {
            error=error+3600;
        }

        if (error<=tol) {

```

```

        motorKiri(servoMundur,100);
        motorKanan(servoMundur,40);
    }
    else if (error>tol) {
        motorKiri(servoMundur,40);
        motorKanan(servoMundur,100);
    }
    else {
        motorKiri(servoMundur,100);
        motorKanan(servoMundur,100);
    }

} while (usKanan1<360);

motorKiri(servoStop,100);
motorKanan(servoStop,100);

}

void mundurKompasSampaiBelakangDekat(void) {
    int sudutTujuan,sudutSkrng,error,tol;

    tol = 20;
    sudutTujuan = bacaKompas();
    motorKiri(servoMundur,100);
    motorKanan(servoMundur,100);

    do {
        delay_ms(50);
        usBlk = read_usound_belakang()*10;

        sudutSkrng = bacaKompas();
        error = sudutTujuan - sudutSkrng;

        if (error>1800)
        {
            error=- (3600-error);
        }
        else if (error <-1800)

```

```

        {
            error=error+3600;
        }

        if (error<=tol) {
            motorKiri(servoMundur,100);
            motorKanan(servoMundur,40);
        }
        else if (error>tol) {
            motorKiri(servoMundur,40);
            motorKanan(servoMundur,100);
        }
        else {
            motorKiri(servoMundur,100);
            motorKanan(servoMundur,100);
        }

    } while (usBlk>280);

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);

}

```

```

void mundurKompasSampaiKiri1Jauh(void) {
    int sudutTujuan,sudutSkr,error,tol;

```

```

    tol = 20;
    sudutTujuan = bacaKompas();
    motorKiri(servoMundur,100);
    motorKanan(servoMundur,100);

    do {
        delay_ms(50);
        usKiri1 = read_usound_kiri1()*10;

        sudutSkr = bacaKompas();
        error = sudutTujuan - sudutSkr;
    } while (error>tol);
}

```

```

        if (error>1800)
        {
            error=- (3600-error);
        }
        else if (error <-1800)
        {
            error=error+3600;
        }

        if (error<=tol) {
            motorKiri(servoMundur,100);
            motorKanan(servoMundur,40);
        }
        else if (error>tol) {
            motorKiri(servoMundur,40);
            motorKanan(servoMundur,100);
        }
        else {
            motorKiri(servoMundur,100);
            motorKanan(servoMundur,100);
        }

    } while (usKiri1<400);

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);
}

void majuKompasSampaiKiri2Jauh(void) {
    int sudutTujuan,sudutSkr,error,tol;

    tol = 20;
    sudutTujuan = bacaKompas();
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {

```

```

        delay_ms(50);
        usKiri2 = read_usound_kiri2()*10;

        sudutSkrng = bacaKompas();
        error = sudutTujuan - sudutSkrng;

        if (error>1800)
        {
            error=- (3600-error);
        }
        else if (error <-1800)
        {
            error=error+3600;
        }

        if (error<=tol) {
            motorKiri(servoMaju,40);
            motorKanan(servoMaju,100);
        }
        else if (error>tol) {
            motorKiri(servoMaju,100);
            motorKanan(servoMaju,40);
        }
        else {
            motorKiri(servoMaju,100);
            motorKanan(servoMaju,100);
        }

    } while (usKiri2<400);

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);
}

void majuKompasSampaiKanan2Jauh(void) {
    int sudutTujuan,sudutSkrng,error,tol;

    lcd_gotoxy(0,1);

```

```

lcd_putsf("mjKompas_Ka_2_jh");

tol = 20;
sudutTujuan = bacaKompas();
motorKiri(servoMaju,100);
motorKanan(servoMaju,100);

do {
    delay_ms(50);
    usKanan2 = read_usound_kanan2()*10;

    sudutSkrng = bacaKompas();
    error = sudutTujuan - sudutSkrng;

    if (error>1800)
    {
        error=- (3600-error);
    }
    else if (error <-1800)
    {
        error=error+3600;
    }

    if (error<=-tol) {
        motorKiri(servoMaju,40);
        motorKanan(servoMaju,100);
    }
    else if (error>tol) {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,40);
    }
    else {
        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);
    }

} while (usKanan2<420);

motorKiri(servoStop,100);
motorKanan(servoStop,100);

```

```

}

void majuKompasSampaiKiri2Dekat(void) {
    int sudutTujuan,sudutSkrng,error,tol;

    tol = 20;
    sudutTujuan = bacaKompas();

    loop_majuKompasSampaiKiri2Dekat:

    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(50);
        usKiri2 = read_usound_kiri2()*10;

        sudutSkrng = bacaKompas();
        error = sudutTujuan - sudutSkrng;

        if (error>1800)
        {
            error=- (3600-error);
        }
        else if (error < -1800)
        {
            error=error+3600;
        }

        if (error< -tol) {
            motorKiri(servoMaju,40);
            motorKanan(servoMaju,100);
        }
        else if (error>tol) {
            motorKiri(servoMaju,100);
            motorKanan(servoMaju,40);
        }
        else {

```

```

        motorKiri(servoMaju,100);
        motorKanan(servoMaju,100);
    }

    } while (usKiri2>300);

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);

    initTimer();
    delay_ms(200);
    usKiri2 = read_usound_kiri2()*10;
    initMotor();
    if ((usKiri2>300)) goto loop_majuKompasSampaiKiri2Dekat;
}

void majuKompasSampaiKanan2Dekat(void) {
    int sudutTujuan,sudutSkrng,error,tol;

    tol = 20;
    sudutTujuan = bacaKompas();

loop_majuKompasSampaiKanan2Dekat:
    motorKiri(servoMaju,100);
    motorKanan(servoMaju,100);

    do {
        delay_ms(50);
        usKanan2 = read_usound_kanan2()*10;

        sudutSkrng = bacaKompas();
        error = sudutTujuan - sudutSkrng;

        if (error>1800)
        {
            error=- (3600-error);
        }
        else if (error <-1800)

```



```

        {
            error=error+3600;
        }

        if (error<-tol) {
            motorKiri(servoMaju,40);
            motorKanan(servoMaju,100);
        }
        else if (error>tol) {
            motorKiri(servoMaju,100);
            motorKanan(servoMaju,40);
        }
        else {
            motorKiri(servoMaju,100);
            motorKanan(servoMaju,100);
        }

    } while (usKanan2>300);

    motorKiri(servoStop,100);
    motorKanan(servoStop,100);

    initTimer();
    delay_ms(200);
    usKanan2 = read_usound_kanan2()*10;
    initMotor();
    if ((usKanan2>300)) goto loop_majuKompasSampaiKanan2Dekat;
}

//void gerakServoCapit(unsigned int periode){
void gerakServoCapit(unsigned int sudut, char pulsa) {

    unsigned int n;
    char i;

    initTimer();

    n = (sudut * 40/9 ) + 200;
    //n= 1150;

```

```

i=0;
while (i<pulsa) {
    servoCapit = 1;
    //delay_us(1500);
    TIFR |= 0x04;
    TCNT1 = 65535 - n;
    TCCR1B=0x02;
    while (!(TIFR & 0x04)) {}
    TCCR1B=0x00;
    servoCapit = 0;
    TIFR |= 0x04;
    //delay_ms(10);
    TCNT1 = 65535 - 15000/2;
    TCCR1B = 0x02;
    while (!(TIFR & 0x04)) {}
    TCCR1B=0x00;
    TIFR |= 0x04;
    i++;
}

initMotor();

//i2c_start();delay_ms(1);    // Start Condition
//i2c_write(0xE0);delay_ms(1); // Tulis ke modul DT-SENSE
//i2c_write(0x06);delay_ms(1);
//i2c_write(periode/256);delay_ms(1);
//i2c_write(periode%256);delay_ms(1);
//i2c_stop();// Stop Condition
//delay_ms(100);
}

void capitBuka(unsigned char pulsa) {

    unsigned char i;

    i=0;
    while (i<pulsa) {
        servoCapit = 1;
        delay_us(300);
        servoCapit = 0;
        delay_ms(10);
    }
}

```

```

        i++;
    }
}

void capiTutup(unsigned char pulsa) {
    unsigned char i;

    i=0;
    while (i<pulsa) {
        servoCapit = 1;
        delay_us(2200);
        servoCapit = 0;
        delay_ms(10);
        i++;
    }
}

void gerakServoLengan(unsigned int sudut, char pulsa)
{
    unsigned int n;
    char i;

    initTimer();

    n = (sudut * 40/9 ) + 200;
    //n= 1150;
    i=0;
    while (i<pulsa) {
        servoLengan = 1;
        //delay_us(1500);
        TIFR |= 0x04;
        TCNT1 = 65535 - n;
        TCCR1B=0x02;
        while (!(TIFR & 0x04)) {}
        TCCR1B=0x00;
        servoLengan = 0;
        TIFR |= 0x04;
        //delay_ms(10);
        TCNT1 = 65535 - 15000/2;
        TCCR1B = 0x02;
    }
}

```

```
        while (!(TIFR & 0x04)) {}
        TCCR1B=0x00;
        TIFR |= 0x04;
        i++;
    }
    initMotor();
}

void angkatBoneka() {

    unsigned char i;

    gerakSensor(120,10);
    i=0;
    while (i<150) {
        servoCapit = 1;
        delay_us(300);
        servoCapit = 0;
        servoLengan = 1;
        delay_us(500);
        servoLengan = 0;
        delay_ms(10);
        i++;
    }

    delay_ms(1000);
    i=0;
    while (i<50) {
        servoCapit = 1;
        delay_us(1000);
        servoCapit = 0;
        delay_ms(10);
        i++;
    }

    i=0;
    while (i<50) {
        servoCapit = 1;
        delay_us(1000);
        servoCapit = 0;
        servoLengan = 1;
        delay_us(1500);
```

```
servoLengan = 0;
delay_ms(10);
i++;
}

i=0;
while (i<50) {
    servoCapit = 1;
    delay_us(700);
    servoCapit = 0;
    servoLengan = 1;
    delay_us(2200);
    servoLengan = 0;
    delay_ms(10);
    i++;
}

}

void taruhBoneka() {

    unsigned char i;

    gerakSensor(120,10);
    i=0;
    while (i<150) {
        servoCapit = 1;
        delay_us(1000);
        servoCapit = 0;
        servoLengan = 1;
        delay_us(500);
        servoLengan = 0;
        delay_ms(10);
        i++;
    }

    delay_ms(1000);
    i=0;
    while (i<50) {
```

```
servoCapit = 1;
delay_us(300);
servoCapit = 0;
delay_ms(10);
i++;
}

i=0;
while (i<50) {
    servoCapit = 1;
    delay_us(300);
    servoCapit = 0;
    servoLengan = 1;
    delay_us(1500);
    servoLengan = 0;
    delay_ms(10);
    i++;
}

i=0;
while (i<50) {
    servoCapit = 1;
    delay_us(300);
    servoCapit = 0;
    servoLengan = 1;
    delay_us(2200);
    servoLengan = 0;
    delay_ms(10);
    i++;
}

}

void bacaSuhu() {
    i2c_start();    // Start Condition
    i2c_write(0xE0); // Tulis ke modul DT-SENSE
    i2c_write(13);
    i2c_stop();    // Stop Condition
```

```

delay_ms(100);

i2c_start();    // Start Condition
i2c_write(0xE1); // Tulis ke modul DT-SENSE

VSensi[0] = i2c_read(1);
VSensi[1] = i2c_read(1);
VSensi[2] = i2c_read(1);
VSensi[3] = i2c_read(1);
VSensi[4] = i2c_read(1);
VSensi[5] = i2c_read(1);
VSensi[6] = i2c_read(1);
VSensi[7] = i2c_read(0);

i2c_stop();    // Stop Condition
}

unsigned char soundActivation(void) {

    unsigned char hasil,high,ketemu;
    int counter,frek,selisih;

    initTimer();
    ketemu = 0;
    frek = 0;
    while(ketemu<7) {
        counter = 0;
        high = 1;
        TIFR |= 0x04;
        TCNT1 = 65535 - 50000; // 100 mS
        TCCR1B=0x02; //
        while (!(TIFR & 0x04)) {
            if (high && PINC.5==0) {
                high = 0;
                counter++;
            }
            if (high==0 && PINC.5==1) {high = 1;}
        }
        TCCR1B=0x00;
    }

```

```

//putchar(counter/10000 %10 +0x30);
//putchar(counter/1000 %10 +0x30);
//putchar(counter/100 %10 +0x30);
//putchar(counter/10 %10 +0x30);
//putchar(counter %10 +0x30);
//putchar(13);

if ((counter>195) && (counter<425)) {
    selisih = frek-counter;
    if ((selisih<3) && (selisih>-3)) ketemu++;
    frek = counter;
}
else {
    ketemu = 0;
    frek = 0;
}
}

frek = frek/10;
if (frek==35) {
    hasil = 0;
}
else if (frek==20) {
    hasil = 1;
}
else if (frek==22) {
    hasil = 2;
}
else if (frek==24) {
    hasil = 3;
}
else if (frek==26) {
    hasil = 4;
}
else if (frek==28) {
    hasil = 5;
}
else if (frek==30) {
    hasil = 6;
}

```



```

    }
    else if (frek==32) {
        hasil = 7;
    }
    else if (frek==34) {
        hasil = 8;
    }
    else if (frek==36) {
        hasil = 9;
    }
    else if (frek==38) {
        hasil = 10;
    }
    else if (frek==40) {
        hasil = 11;
    }
    else if (frek==42) {
        hasil = 12;
    }

    initMotor();

    return hasil;
}

char cariUVApi(unsigned int sudutAwal, unsigned int sudutAkhir) {
    unsigned int tunda;
    unsigned char adaApi,temp;
    initTimer();
    gerakSensor(sudutAwal,50);

    sudut = sudutAwal;
    adaApi = 0;
    temp = 0;
    while ((sudut<sudutAkhir) && (adaApi<2)) {
        delay_ms(1);
        tunda = 0;
        //while (tunda<1000) {
        //    if (uvTronPIN) adaApi++;
        //    tunda++;
        //}
    }
}

```

```

        //if ( temp == adaApi) adaApi=0;
        if (adaApi==0) {
            sudut = sudut + 3;
            gerakSensor(sudut,5);
        }
        temp = adaApi;
    }

    initMotor();
    return adaApi;
}

char cariIRApi(unsigned int sudutAwal, unsigned int sudutAkhir) {
    unsigned char adaApi,i;

    initTimer();
    gerakSensor(sudutAwal,50);

    sudut = sudutAwal;
    adaApi = 0;
    while ((sudut<sudutAkhir) && (adaApi<1)) {
        bacaSuhu();

        for (i=0;i<8;i++) {

            //putchar(VSensi[i]/100 %10 + 0x30);
            //putchar(VSensi[i]/10 %10 + 0x30);
            //putchar(VSensi[i] %10 + 0x30);
            //putchar(0x20);
            if (VSensi[i]>50) {

                //bacaSuhu();

                for (i=0;i<8;i++) {

                    if (VSensi[i]>50) {

                        buzzer = 1;
                        delay_ms(200);
                        buzzer = 0;
                        adaApi = 1;

```

```

        break;
    }
}
}
//putchar(0x0d);
sudut = sudut + 2;
gerakSensor(sudut,5);
}

initMotor();
return adaApi;
}

char cariIRApiSerong(unsigned int sudutAwal, unsigned int sudutAkhir) {
    unsigned char adaApi,i;

    initTimer();
    gerakSensor(sudutAwal,50);

    sudut = sudutAwal;
    adaApi = 0;
    while ((sudut<sudutAkhir) && (adaApi<1)) {
        bacaSuhu();

        for (i=0;i<8;i++) {

            //putchar(VSensi[i]/100 %10 + 0x30);
            //putchar(VSensi[i]/10 %10 + 0x30);
            //putchar(VSensi[i] %10 + 0x30);
            //putchar(0x20);
            if (VSensi[i]>40) {

                //bacaSuhu();

                for (i=0;i<8;i++) {

                    if (VSensi[i]>40) {

                        buzzer = 1;

```

```

        delay_ms(200);
        buzzer = 0;
        adaApi = 1;
        break;
    }
}
}
//putchar(0x0d);
sudut = sudut + 2;
gerakSensor(sudut,5);
}

initMotor();
return adaApi;
}

```

```

char cariPosisiApiIR(char batas1, char batas2)
{
    unsigned int sudut;
    char adaApi=0;

    initTimer();

    gerakSensor(batas1,20);
    sudut = batas1;

    while(1)
    {
        if (sudut>=batas2) break;
        sudut = sudut + 5;
        gerakSensor(sudut,10);

        if (photo1==0 || photo2==0 )
        {
            adaApi = 1;
            break;
        }
    }
    initMotor();
    return adaApi;
}

```

}

```
void tiupLilin(void) {  
    initTimer();  
    kipas = 1;  
    gerakSensor(0,50);  
    gerakSensor(90,50);  
    gerakSensor(150,50);  
    gerakSensor(90,50);  
    gerakSensor(0,50);  
    kipas = 0;  
  
    initMotor();  
}
```

```
void angkatBendera(void) {  
    //jatuhkan beeper  
    delay_ms(500);  
    bendera = 1;  
    delay_ms(30);  
    bendera = 0;  
    delay_ms(500);  
    bendera = 1;  
    delay_ms(30);  
    bendera = 0;  
    delay_ms(500);  
    bendera = 1;  
    delay_ms(30);  
    bendera = 0;  
}
```

```
void tiupLilinDonk(){  
  
    kipas = 1;  
  
    gerakSensor(90,20);  
    gerakSensor(50,20);  
    gerakSensor(90,20);
```

```
        gerakSensor(160,20);
        gerakSensor(90,20);
        gerakSensor(50,20);
        gerakSensor(90,20);
        gerakSensor(160,20);
        kipas = 0;

        gerakSensor(210,50);
    }

    void main(void)
    {
        // Declare your local variables here+
        unsigned char temp,temp1;
        unsigned int sudutStart;

        initIO();
        i2c_init();
        initMotor();

        // Global enable interrupts
        #asm("sei")

        beep(2);

        // sudut di WM
        atas = 410;
        bawah = 2597;
        kanan = 952;
        kiri = 3356;

        // sudut di its
        atas = 1750;
        bawah = 0;
        kanan = 2700;
        kiri = 1000;

        kor = 100; // 50
```

```

// LCD module initialization
lcd_init(16);

lcd_clear();
lcd_putsf("-- MERCURY --");

motorKiri(servoStop,100);
motorKanan(servoStop,100);

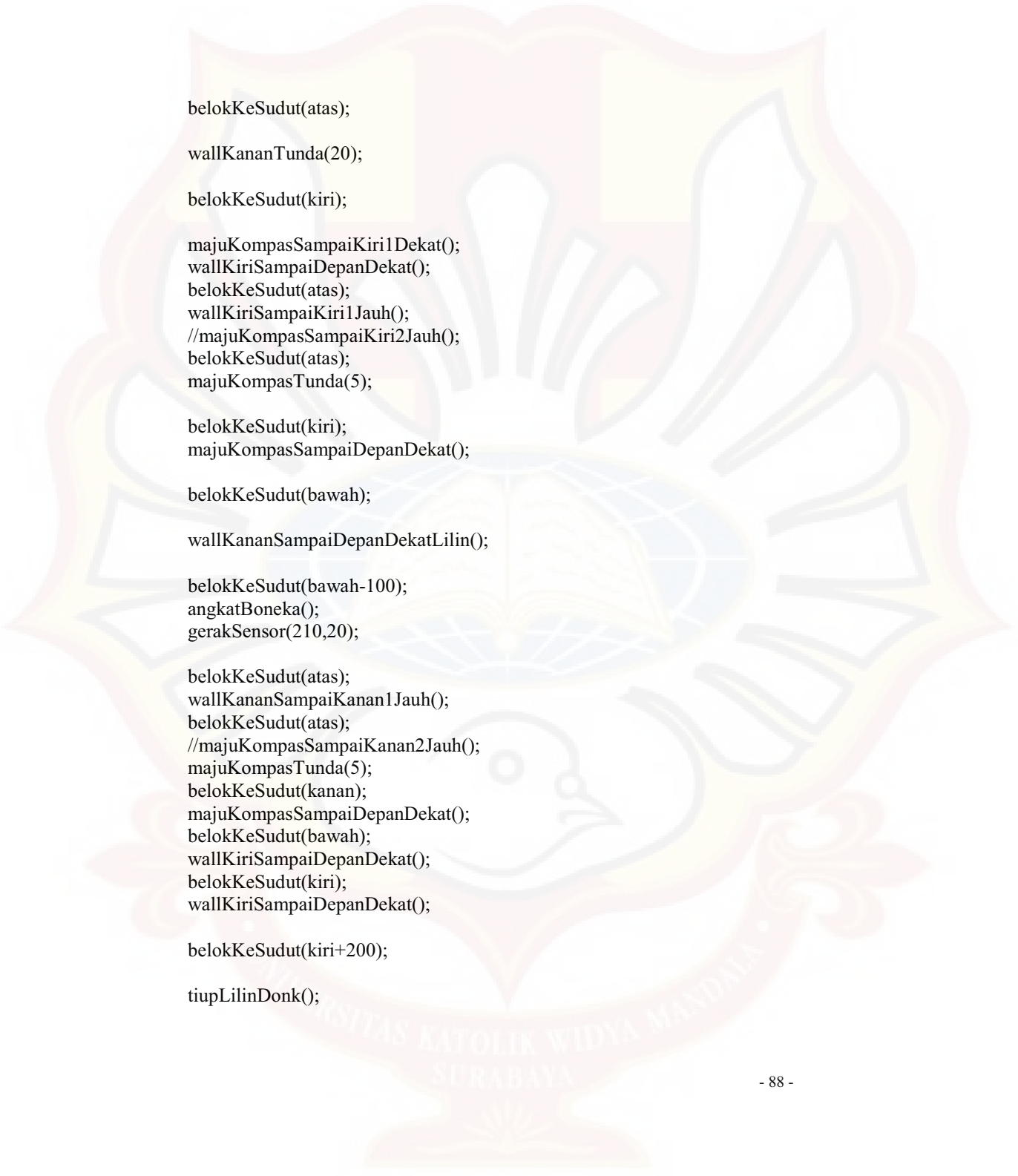
//cekKompas();
//motorKiri(servoMaju,100);
//motorKanan(servoMaju,100);

//cekUS();
//while(1);

//delay_ms(1000);
//capitBuka(50);
//gerakSensor(210,50);


if (micPin) {
beep(1);
////////// start merah
belokKeSudut(kanan);
majuKompasSampaiDepanDekat();
belokKeSudut(atas);
majuKompasSampaiDepanDekat();
//////////
//belokKeSudut(kanan);
//wallKiriSampaiKiri1Jauh();
belokKeSudut(kanan);
majuKompasSampaiKiri2Jauh();
belokKeSudut(kanan);
majuKompasSampaiKiri1Dekat();
//belokKeSudut(kanan);
wallKiriSampaiKiri1Jauh();
belokKeSudut(kanan);
majuKompasSampaiDepanDekat();
//////////

```



```
belokKeSudut(atas);  
  
wallKananTunda(20);  
  
belokKeSudut(kiri);  
  
majuKompasSampaiKiri1Dekat();  
wallKiriSampaiDepanDekat();  
belokKeSudut(atas);  
wallKiriSampaiKiri1Jauh();  
//majuKompasSampaiKiri2Jauh();  
belokKeSudut(atas);  
majuKompasTunda(5);  
  
belokKeSudut(kiri);  
majuKompasSampaiDepanDekat();  
  
belokKeSudut(bawah);  
  
wallKananSampaiDepanDekatLilin();  
  
belokKeSudut(bawah-100);  
angkatBoneka();  
gerakSensor(210,20);  
  
belokKeSudut(atas);  
wallKananSampaiKanan1Jauh();  
belokKeSudut(atas);  
//majuKompasSampaiKanan2Jauh();  
majuKompasTunda(5);  
belokKeSudut(kanan);  
majuKompasSampaiDepanDekat();  
belokKeSudut(bawah);  
wallKiriSampaiDepanDekat();  
belokKeSudut(kiri);  
wallKiriSampaiDepanDekat();  
  
belokKeSudut(kiri+200);  
  
tiupLilinDonk();
```





```

belokKeSudut(kanan);
wallKananSampaiDepanDekat();
belokKeSudut(atas);

majuKompasSampaiDepanDekat();
//////////
//belokKeSudut(kanan);
wallKiriSampaiKiri1Jauh();
belokKeSudut(kanan);
majuKompasSampaiKiri2Jauh();

//////////
belokKeSudut(atas);
majuKompasTunda(10);
belokKeSudut(kiri);
majuKompasSampaiKiri1Dekat();
belokKeSudut(kiri);
wallKiriSampaiKiri1Jauh();

belokKeSudut(kiri);
majuKompasSampaiKiri1Dekat();
belokKeSudut(kiri);
wallKiriSampaiKiri1Jauh();

belokKeSudut(kiri);
majuKompasSampaiKiri1Dekat();
belokKeSudut(kiri);
wallKiriSampaiDepanDekat();

//////////
belokKeSudut(atas);
wallKiriSampaiDepanDekat();
belokKeSudut(kanan);
wallKiriSampaiDepanDekatLilin();

belokKeSudut(kanan-200);
tiupLilinDonk();

```

////////////////

belokKeSudut(kiri);  
wallKananSampaiDepanDekat();  
belokKeSudut(bawah);  
wallKananSampaiDepanDekat();  
belokKeSudut(kanan);  
wallKananSampaiKanan1Jauh();  
belokKeSudut(kanan);  
//majuKompasSampaiKanan2Jauh();  
majuKompasTunda(5);  
belokKeSudut(bawah);  
majuKompasSampaiDepanDekat();  
belokKeSudut(kiri);

majuKompasTunda(4);

taruhBoneka();

////////////////////////////////////

belokKeSudut(kanan);  
wallKananSampaiDepanDekat();  
belokKeSudut(atas);  
majuKompasSampaiDepanDekat();  
belokKeSudut(kanan);  
wallKananSampaiKanan1Jauh();  
belokKeSudut(kanan);  
//majuKompasSampaiKanan2Jauh();  
majuKompasTunda(5);  
belokKeSudut(bawah);  
majuKompasSampaiDepanDekat();  
belokKeSudut(kiri);  
wallKiriSampaiDepanDekat();

belokKeSudut(kiri+200);

tiupLilinDonk();

belokKeSudut(kanan);  
wallKananSampaiDepanDekat();  
belokKeSudut(atas);  
wallKananSampaiKanan1Jauh();

```
belokKeSudut(atas);  
majuKompasSampaiKanan1Dekat();  
belokKeSudut(atas);  
majuKompasTunda(4);  
belokKeSudut(kiri);  
majuKompasSampaiDepanDekat();  
belokKeSudut(atas);  
wallKiriSampaiDepanDekat();
```

```
belokKeSudut(atas-200);  
tiupLilinDonk();
```

```
belokKeSudut(kanan);  
majuKompasSampaiDepanDekat();  
belokKeSudut(atas);  
wallKananSampaiKanan1Jauh();  
belokKeSudut(bawah);  
wallKiriTunda(15);  
belokKeSudut(kiri);  
majuKompasSampaiDepanDekat();  
belokKeSudut(bawah);  
wallKananSampaiDepanDekat();
```

```
belokKeSudut(bawah+200);  
tiupLilinDonk();
```

```
while(1);  
  
}  
else {  
beep(2);  
////////// start biru  
belokKeSudut(kanan);  
majuKompasSampaiDepanDekat();  
belokKeSudut(bawah);  
majuKompasSampaiDepanDekat();  
//////////  
//belokKeSudut(kanan);  
//wallKananSampaiKanan1Jauh();
```



```
belokKeSudut(kanan);
majuKompasSampaiKanan2Jauh();
belokKeSudut(kanan);
majuKompasSampaiKanan1Dekat();
//belokKeSudut(kanan);
wallKananSampaiKanan1Jauh();
belokKeSudut(kanan);
majuKompasSampaiDepanDekat();
//////////
```

```
belokKeSudut(bawah);
```

```
wallKiriTunda(20);
```

```
belokKeSudut(kiri);
```

```
majuKompasSampaiKanan1Dekat();
wallKananSampaiDepanDekat();
belokKeSudut(bawah);
wallKiriSampaiKiri1Jauh();
//majuKompasSampaiKiri2Jauh();
belokKeSudut(bawah);
majuKompasTunda(5);
```

```
belokKeSudut(kiri);
majuKompasSampaiDepanDekat();
```

```
belokKeSudut(atas);
```

```
wallKiriSampaiDepanDekatLilin();
```

```
belokKeSudut(atas+100);
angkatBoneka();
gerakSensor(210,20);
```

```
belokKeSudut(bawah);
wallKananSampaiKanan1Jauh();
belokKeSudut(bawah);
//majuKompasSampaiKanan2Jauh();
majuKompasTunda(5);
belokKeSudut(kanan);
```

majuKompasSampaiDepanDekat();  
belokKeSudut(atas);  
wallKananSampaiDepanDekat();  
belokKeSudut(kiri);  
wallKananSampaiDepanDekat();

belokKeSudut(kiri-200);

tiupLilinDonk();

belokKeSudut(kanan);  
wallKiriSampaiDepanDekat();

//////////

belokKeSudut(bawah);  
majuKompasTunda(10);  
belokKeSudut(kiri);  
majuKompasSampaiKanan1Dekat();  
belokKeSudut(kiri);  
wallKiriSampaiKiri1Jauh();

belokKeSudut(kiri);  
majuKompasSampaiKanan1Dekat();  
belokKeSudut(kiri);  
wallKiriSampaiKiri1Jauh();

belokKeSudut(kiri);  
majuKompasSampaiKanan1Dekat();  
belokKeSudut(kiri);  
wallKananSampaiDepanDekat();

////////

belokKeSudut(bawah);  
wallKananSampaiDepanDekat();  
belokKeSudut(kanan);  
wallKananSampaiDepanDekatLilin();

belokKeSudut(kanan+200);  
tiupLilinDonk();

```

////////////////
belokKeSudut(kiri);
wallKiriSampaiDepanDekat();
belokKeSudut(atas);
wallKiriSampaiDepanDekat();
belokKeSudut(kanan);
wallKiriSampaiKiri1Jauh();
belokKeSudut(kanan);
//majuKompasSampaiKanan2Jauh();
majuKompasTunda(5);
belokKeSudut(atas);
majuKompasSampaiDepanDekat();
belokKeSudut(kiri);

```

```

majuKompasTunda(4);

```

```

taruhBoneka();

```

```

////////////////
belokKeSudut(kanan);
wallKiriSampaiDepanDekat();
belokKeSudut(bawah);
majuKompasSampaiDepanDekat();
belokKeSudut(kanan);
wallKiriSampaiKiri1Jauh();
belokKeSudut(kanan);
//majuKompasSampaiKanan2Jauh();
majuKompasTunda(5);
belokKeSudut(atas);
majuKompasSampaiDepanDekat();
belokKeSudut(kiri);
wallKiriSampaiDepanDekat();

```

```

belokKeSudut(kiri-200);

```

```

tiupLilinDonk();

```

```

belokKeSudut(kanan);
wallKiriSampaiDepanDekat();
belokKeSudut(bawah);

```

```
wallKiriSampaiKiri1Jauh();  
belokKeSudut(bawah);  
majuKompasSampaiKiri1Dekat();  
belokKeSudut(bawah);  
majuKompasTunda(4);  
belokKeSudut(kiri);  
majuKompasSampaiDepanDekat();  
belokKeSudut(bawah);  
wallKananSampaiDepanDekat();
```

```
belokKeSudut(bawah+200);  
tiupLilinDonk();
```

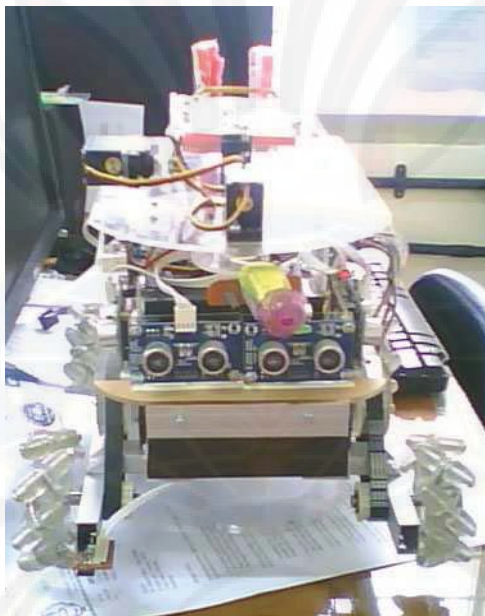
```
belokKeSudut(kanan);  
majuKompasSampaiDepanDekat();  
belokKeSudut(bawah);  
wallKiriSampaiKiri1Jauh();  
belokKeSudut(atas);  
wallKiriTunda(15);  
belokKeSudut(kiri);  
majuKompasSampaiDepanDekat();  
belokKeSudut(atas);  
wallKiriSampaiDepanDekat();
```

```
belokKeSudut(atas-200);  
tiupLilinDonk();
```

```
while(1);  
}
```

```
while (1)  
{  
    // Place your code here  
  
    //delay_ms(2000);  
  
};  
}
```

**LAMPIRAN 6**  
**FOTO ROBOT**

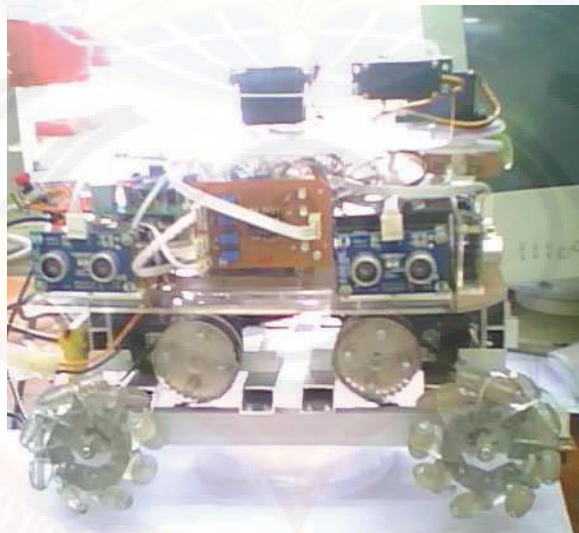


**Robot Tampak Depan**





**Robot Tampak Belakang**



**Robot Tampak Samping**



**Robot Tampak Bawah**



**Robot Tampak Sudut Kiri**

## LAMPIRAN 7

### PERATURAN KRCI 2009

*Rule Book of Expert Battle Robocon 2009*

#### KONTES ROBOT CERDAS INDONESIA

### Divisi Expert Battle

#### I PENDAHULUAN

KRCI (Kontes Robot Cerdas Indonesia) selama ini menerapkan aturan kontes atau pertandingan dalam bentuk perlombaan yang bersifat individual. Hal ini menyebabkan pertandingan relatif menjadi kurang begitu atraktif bagi penonton, karena hanya peserta, juri dan pengamat teknologi saja yang dapat memahami apa yang sedang dilakukan oleh robot ketika bertanding. Sedangkan pada KRI (Kontes Robot Indonesia), pertandingan menjadi meriah sebab robot-robot peserta saling berhadap-hadapan seperti layaknya pertandingan olahraga sehingga memungkinkan adanya dukungan aktif para penonton/supporter ketika pertandingan sedang berlangsung. Sementara itu, jika robot KRCI selama ini relatif berbentuk kecil sehingga mudah dibawa, maka robot KRI hampir selalu berbentuk relatif besar dan berjumlah lebih dari dua. Dewan Juri dan DP2M DIKTI menyadari bahwa banyak perguruan tinggi belum mampu mengikuti kontes tipe KRI karena terkendala dengan persiapan robot yang besar-besar dan latihan yang membutuhkan lapangan berdimensi besar juga.

Untuk itulah maka dalam KRCI 2009, sebuah divisi baru bernama Expert Battle diperkenalkan. Konsep pertandingan disusun dengan menggabungkan konsep KRI yang mensyaratkan sistem pertandingan berhadap-hadapan dengan KRCI yang melombakan robot berdimensi kecil yang berfungsi – secara sendiri-sendiri – untuk memadamkan api lilin dan menyelamatkan bayi. Karena sifatnya yang berhadap-hadapan tersebut maka divisi baru ini, diberi nama Expert Battle.

Divisi Expert Battle ini sesungguhnya mengadopsi peraturan pertandingan IJE Robocon (Indonesia Japan Expo) Robot Contest 2008 yang telah diselenggarakan pada 9 Nopember 2008 yang lalu dalam rangka ulang tahun ke-50 persahabatan Indonesia-Jepang.

#### II. TEMA

Tema Divisi Pemadam Api Divisi Expert Battle pada KRCI 2009 adalah “API-PERSAHABATAN”

#### III. RULE OF THE GAME

Dua robot autonomous (otomatis) akan saling berhadap-hadapan dalam pertandingan ini untuk memperebutkan posisi-posisi api dan memadamkannya dalam waktu 3 menit dan menyelamatkan boneka bayi dengan mengangkatnya dan memindahkannya ke posisi HOME masing-masing. Robot yang paling banyak

memadamkan api – yang dihitung berdasarkan skor posisi api yang berhasil dipadamkan dengan sah – dan menyelamatkan boneka bayi akan menjadi pemenang. Pemenang juga ditentukan dari siapa yang paling cepat (dalam rentang 3 menit) memadamkan lilin-lilin pada posisi tertentu dan memindah boneka bayi ke HOME masing-masing (dijelaskan di bab-bab berikut).

### 3.1 Tim Robot

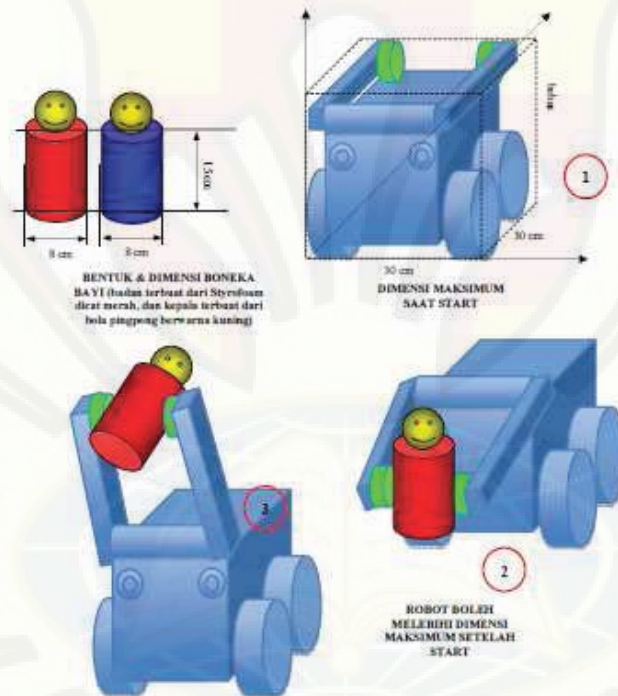
Tim Robot harus berasal dari Perguruan Tinggi yang terdiri dari 2 (dua) mahasiswa aktif dan 1 (satu) pembimbing.

### 3.2 Spesifikasi Robot

Robot yang dipertandingkan harus memenuhi spesifikasi sebagai berikut:

- a) Robot hanya 1 (satu) unit yang boleh bertanding dan memiliki dimensi panjang-lebar maksimum (30x30)cm ketika berada dalam posisi START di HOME POSITION dengan berat dan tinggi tidak dibatasi, kecuali untuk tinggi kipas atau mulut blower pemadam api tingginya dibatasi tidak boleh melebihi 25 cm diukur dari lantai.
- b) Robot boleh merubah dimensinya melebihi ukuran pada (a) selama pertandingan tetapi tinggi kipas atau mulut blower pemadam api tetap tidak boleh melebihi 25 cm.
- c) Robot tidak boleh melakukan separasi (pemisahan) menjadi lebih dari satu robot ataupun melepas bagian robotnya di lapangan pertandingan.
- d) Sistem pemadam api (lilin) tidak boleh menggunakan cairan, gas dan semacannya. Sistem yang diperkenankan hanya yang menggunakan semburan udara kering seperti kipas atau pneumatik.
- e) Sistem pengangkat boneka bayi boleh menggunakan teknik-teknik robot tangan atau teknik lain yang mungkin.
- f) Robot boleh menggunakan berbagai teknik gerak yang mungkin, termasuk sistem roda atau kaki.
- g) Bentuk robot ketika START dan bertanding divisualisasikan dalam Gambar 1 berikut ini.
- h) Besar Tegangan yang digunakan di rangkaian elektronik dan motor robot dibatasi maksimum 24 Volt.
- i) Robot harus dapat berjalan sendiri secara autonomous tanpa perintah dari luar, dan oleh karena itu dilarang menggunakan sistem RF (radio frequency) seperti bluetooth, sistem komunikasi optik, wifi dan semacannya.
- j) Sistem Power Supply robot HARUS di-ON-kan dengan menggunakan hanya SATU SWITCH saja.
- k) Sistem START robot HARUS menggunakan SATU TOMBOL PUSH BOTTON yang berada di tubuh robot dan mudah dijangkau. Tombol lain yang diperkenankan ada hanyalah TOMBOL RESET.

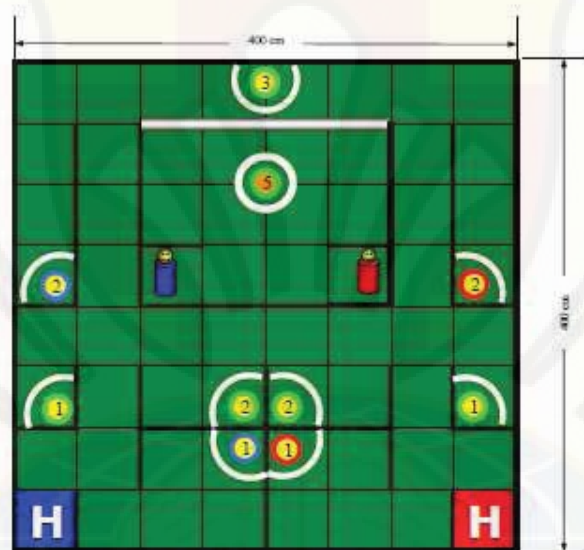




Gambar 1: Bentuk Robot ketika START, bertanding dan mengangkat boneka bayi

### 3.3 Lapangan Pertandingan

Lapangan Pertandingan ditunjukkan dalam Gambar 1 berikut ini.

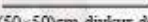


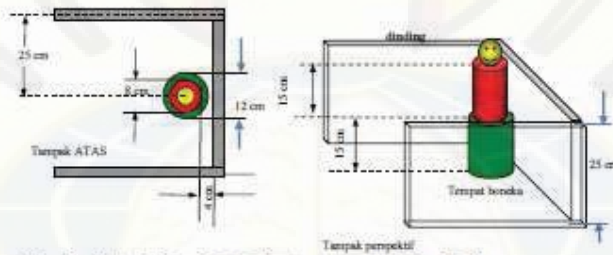
Gambar 2: Lapangan Pertandingan Divisi Pemadam Api Divisi Expert Battle


Tabel 1: Keterangan Gambar 2

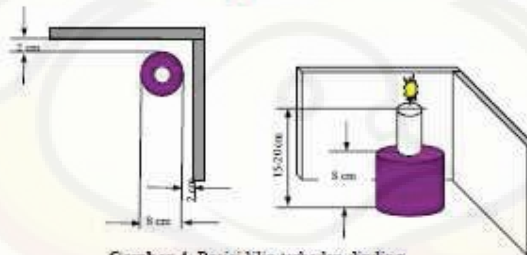
|  |  |  |   |
|--|--|--|---|
|  | Posisi titik bernomor 1. Nilai Batal jika robot menyentuh titik/tempatnya dan dikuras pemadli 1.   |  | Posisi titik bernomor 5 dengan garis lingkaran putih (d=40cm, tebal=4cm).   |
|  | Posisi titik bernomor 2. Nilai Batal jika robot menyentuh titik/tempatnya dan dikuras pemadli 2.   |  | Akan diberikan nilai 5 jika robot menyentuh titik/tempatnya.  |
|  | Posisi titik bernomor 2. Nilai Batal jika robot menyentuh titik/tempatnya dan dikuras pemadli 2.   |  | Robot di MERAH MENANG MUTI-AR, jika memadamkan titik-titik di posisi ini dan memadamkan semua titik MERAH ke posisi HOME MERAH. |
|  | Posisi titik bernomor 3. Nilai Batal jika robot menyentuh titik/tempatnya dan dikuras pemadli 3.   |  | Robot di BIRU MENANG MUTI-AR, jika memadamkan titik-titik di posisi ini dan memadamkan semua titik BIRU ke posisi HOME BIRU.    |
|  | Lingkaran/kurva garis putih berjari-jari 30cm dan setebal 4cm yang melingkupi tiap pos. titik sebagai petunjuk peredaran api titik secara sah (robot harus menyentuh atau masuk garis melingkari titik ketika memadamkan api). |  | Obstacle berbentuk silinder setengah lingkaran berwarna PUTIH di lantai, dipasang tetap dan tidak dapat dipindah.               |

**Keterangan Tambahan:**

- Lantai lapangan berwarna hijau gelap.
- Lantai lapangan adalah RATA tanpa halangan (obstacle), kecuali pada posisi yang bertanda gambar .
- Tiap kotak berukuran (50x50)cm diukur dari tengah dinding setebal 1.8-2 cm setinggi 25cm.
- Sisi dinding berwarna putih.
- Atas dinding berwarna hitam.
- Semua dinding rata tanpa asesori.
- Tanda H adalah Posisi START Robot di MERAH ataupun BIRU.
- Lambang H berwarna putih.
- Batas tiap kotak (50x50)cm tidak bergaris.
- Posisi Boneka Bayi terhadap dinding adalah seperti Gambar 3 berikut ini:

**Gambar 3:** Posisi Boneka Bayi dan tempatnya terhadap dinding

- Boneka diletakkan di atas tempat berbentuk silinder kayu pejal berwarna hijau yang ditanam tetap di lantai (ukuran: perhatikan Gambar 3)
- Posisi lilin terhadap dinding pojok adalah seperti Gambar 4 berikut (kecuali untuk lilin pada posisi gambar  jaraknya adalah 2 cm dari dinding terdekat):

**Gambar 4:** Posisi lilin terhadap dinding

**Keterangan Gambar 4:**

- Tempat lilin ditanam tetap (kuat) di lantai dengan cara disekrup, terbuat dari kayu.
- Lilin diletakkan di atas kayu tertanam di lubang tengahnya sedalam lebih dari 2 cm, lilin mudah patah jika dinubruk langsung.

**3.4 Sistem Pertandingan**

Pertandingan dilaksanakan sebagai berikut:

- Pertandingan dilakukan dengan Sistem Round Robin (Kompetisi) pada babak Penyisihan. Babak Babak Perempat Final, Semi Final dan Final dilakukan dengan Sistem Gugur (*Knock Out System*).
- Dua Robot akan berhadapan (START di tempat masing-masing) untuk memerebutkan posisi-posisi lilin dan memadamkannya, dan memindah boneka bayi yang menjadi tanggungjawabnya (berwarna sesuai dengan posisi START) ke posisi HOME masing-masing.
- Diberikan waktu maksimum 1 (satu) menit kepada Tim Robot untuk mempersiapkan robotnya di posisi START.
- Robot dijalankan dengan menekan SATU TOMBOL START setelah tanda GO diberikan. Penekanan tombol boleh dilakukan oleh anggota tim.
- Robot akan dinyatakan MENANG MUTLAK jika berhasil memadamkan pasangan lilin sesuai dengan warna tim dan memindah boneka bayi ke posisi HOME masing-masing seperti pada Tabel 1. Nilai totalnya akan ditambah dengan nilai lilin-lilin yang belum padam di seluruh lapangan.
- Jika tidak ada yang MENANG MUTLAK hingga 3 menit pertandingan berakhir maka nilai total dihitung dari lilin-lilin yang berhasil dipadamkan dan keberhasilan mengangkat dan atau memindah boneka bayi ke posisi HOME setelah dikurangi PENALTI.
- WASIT berhak menghentikan pertandingan sewaktu-waktu jika robot membahayakan lapangan, misalnya merobohkan dinding, merusak boneka bayi, menggulingkan lilin sehingga memungkinkan terjadinya kebakaran, dan hal-hal lain yang dianggap dapat membahayakan/merusak lapangan pertandingan.

**3.5 Sistem Penilaian**

- Robot memperoleh nilai 5 jika mampu memadamkan lilin di posisi bertanda angka 5 dalam jarak kurang dari 20 cm diukur dari titik tengah lilin (robot telah menyentuh atau berada di dalam garis pembatas). Nilai batal jika robot masih berada di luar garis, dan atau menyentuh tempat lilin dan atau lilin ini sekaligus mendapat pengurangan nilai (penalti) 5.
- Robot memperoleh nilai 3 jika mampu memadamkan lilin di posisi bertanda angka 3 dalam jarak kurang dari 30 cm diukur dari titik tengah lilin (robot telah menyentuh atau berada di dalam garis pembatas). Nilai batal jika robot masih



berada di luar garis, dan atau menyentuh tempat lilin dan atau lilin ini sekaligus mendapat pengurangan nilai (penalti) 3.

- c. Robot memperoleh nilai 2 jika mampu memadamkan lilin di posisi bertanda angka 2 dalam jarak kurang dari 30 cm diukur dari titik tengah lilin (robot telah menyentuh atau berada di dalam garis pembatas). Nilai batal jika robot masih berada di luar garis, dan atau menyentuh tempat lilin dan atau lilin ini sekaligus mendapat pengurangan nilai (penalti) 2.
- d. Robot memperoleh nilai 1 jika mampu memadamkan lilin di posisi bertanda angka 1 dalam jarak kurang dari 30 cm diukur dari titik tengah lilin (robot telah menyentuh atau berada di dalam garis pembatas). Nilai batal jika robot masih berada di luar garis, dan atau menyentuh tempat lilin dan atau lilin ini sekaligus mendapat pengurangan nilai (penalti) 1.
- e. Robot memperoleh nilai 4 jika berhasil mengangkat sebuah boneka bayi, baik boneka MERAH maupun BIRU dan mempertahankannya di udara selama pertandingan berlangsung. Mendapat nilai 8 jika dua-duanya mampu diangkat dan dipertahankan di udara selama pertandingan.
- f. Robot memperoleh nilai 8 jika mampu mengangkat boneka bayi sesuai warna HOME-nya dan meletakkannya di posisi HOME-nya sendiri (nilai 4 untuk mengangkat dan nilai 4 untuk meletakkan).
- g. Jika sebuah lilin dipadamkan secara bersama-sama oleh dua robot maka nilainya akan dibagi dua.
- h. Nilai total akhir sebuah pertandingan dihitung sesuai dengan aturan pada 3.4.e dan 3.4.f.

### 3.6 Retry (mengulang START)

Izin mengulang START Robot (Retry) diberikan dengan syarat:

- a. Untuk masing-masing Tim, Retry hanya dilakukan sekali selama satu pertandingan. Permintaan Retry harus diajukan secara lisan kepada Wasit, dan baru dilakukan jika IJIN telah diberikan. Peserta boleh melakukan sendiri pemindahan robot ke posisi START ataupun minta bantuan wasit jika perlu.
- b. Boneka bayi yang dirubruk atau disentuh oleh robot akan dikembalikan pada posisi semula jika tim robot ini minta RETRY.
- c. Permintaan Retry TIDAK AKAN DILAYANI jika pertandingan telah berjalan lebih dari SATU MENIT.

### 3.7 Menyerah

Menyerah atau GIVE UP dari sebuah pertandingan dapat diminta dengan prosedur:

- a. Melakukan permintaan secara lisan kepada Wasit sebelum atau ketika pertandingan sedang berlangsung.

- b. Jika permintaan GIVE UP dilakukan ketika pertandingan sedang berlangsung maka robot yang bersangkutan harus DIANGKAT KELUAR dari arena pertandingan.

### 3.8 Penalti

Penalti akan diberikan kepada Tim Robot jika:

- a. Robot menyentuh lilin dan atau tempatnya. Pengurangan nilai DISESUAIKAN dengan NILAI POSISI lilin tersebut.
- b. Robot dengan sengaja menjatuhkan boneka bayi lawan baik langsung (dengan menyenggol) ataupun tak langsung (dengan meniup), kecuali robot memang berusaha mengambil boneka namun terlepas. Nilai Penalti untuk pelanggaran ini adalah 4.
- c. Jika lilin ditubruk dan lepas dari tempatnya maka nilai penalti dikalikan 2 (dua).
- d. Pengurangan nilai SEBESAR 5 akan diberikan kepada Tim yang robotnya menabrak robot lawan hingga menggulingkannya.

Kejadian-kejadian yang TIDAK MENYEBABKAN Penalti:

- a. Robot bertabrakan, baik SENGAJA ATAUPUN TIDAK dan TIDAK MENYEBABKAN robot lawan TERGULING.
- b. Robot terguling dengan sendirinya.
- c. Robot terbakar atau rusak karena alasan sendiri ataupun karena tabrakan.

### 3.9 Diskualifikasi

Diskualifikasi diberikan kepada Tim Robot jika:

- a. Robot tidak memenuhi spesifikasi seperti yang diterangkan dalam Rule 3.2.
- b. Robot merusak lapangan pertandingan seperti, merobokkan dinding, merusak boneka, dan membruk pot lilin dan atau pot boneka hingga lepas dari tempatnya.
- c. Anggota Tim dengan sengaja menyentuh robot ketika sedang bertanding.
- d. Anggota Tim tidak patuh pada arahan WASIT dan atau JURI.
- e. Anggota Tim melakukan tindakan yang dapat dikategorikan sebagai tindakan menodai spirit fair play selama kegiatan kontes.

## IV. PENGHARGAAN

Penghargaan diberikan kepada Tim untuk Kategori:

- a. Champion (Grand Prix) sebagai pemenang pertama.
- b. First Runner Up sebagai pemenang kedua.

- c. Second Runner Up sebagai pemenang ketiga.
- d. The Best Idea sebagai Tim dengan Ide Terbaik
- e. The Best Design sebagai Tim dengan Desain Terbaik.

Penghargaan akan diberikan dalam bentuk Piala, Sertifikat dan Hadiah Khusus yang akan ditentukan kemudian.

#### V. INFORMASI TAMBAHAN dan FAQ (FREQUENTLY ASK QUESTIONS)

Informasi Tambahan dan kolom FAQ akan diberikan sesuai dengan kebutuhan hingga menuju hari pertandingan.

##### Penjelasan Peraturan dan Amandemen:

- 5.1 Robot diperkenankan berubah bentuknya melebihi dimensi pada saat START namun dilarang melebihi batas dinding yang melingkupinya. Tetapi bagaimanapun juga tinggi kipas (atau sisi kipas ketika berputar) dan atau mulut blower pemadam api tidak boleh melebihi 25 cm selama pertandingan.
- 5.2 Robot dilarang memanjat dinding dan atau menjulurkan bagian tubuhnya melebihi batas dinding secara vertikal. Pelanggaran atas hal ini akan menyebabkan penalti sebesar 5 pada 10 detik pertama, dan penalti 3 pada tiap 10 detik berikutnya.
- 5.3 Jika robot mengalami kejadian 5.2 dan robot masih memiliki kesempatan untuk RETRY maka tim robot dapat meminta RETRY. Dalam hal ini prosedur RETRY tetap mengikuti peraturan 3.6.
- 5.4 Jika dalam kasus 5.3 tim robot tidak meminta RETRY hingga 1 menit pertandingan berlangsung maka robot DIANGGAP MENYERAH dan robot akan diangkat keluar dari arena pertandingan.
- 5.5 Jika dalam kasus 5.3 & 5.4 robot bersentuhan/tabrakan dengan robot lawan hingga tidak mudah dipisahkan maka WASIT dan atau JURI akan mengambil tindakan yang diperlukan, misalnya meminta tim robot untuk mematikan sementara robot-robotnya hingga salah satu (robot yang melanggar peraturan 5.1 – 5.4) diangkat dari arena pertandingan. Sedangkan robot yang menjadi korban pelanggaran dapat meneruskan kembali perjalanannya (RUN) dengan meng-ON-kan kembali POWER SWITCH dan atau RESET pada posisi terakhir ketika tabrakan terjadi.
- 5.6 Hasil aktifitas robot pada kasus pelanggaran 5.1 dan atau 5.2 tidak akan dihitung dalam nilai total, misalnya lilin yang dipadamkan ataupun boneka bayi yang diangkat.
- 5.7 Lilin yang padam karena kasus pelanggaran 5.1 dan atau 5.2 akan akan dibiarkan padam dan robot yang menyebabkan padamnya lilin ini akan



dikenai penalti sesuai dengan skor lilin yang bersangkutan. Lilin yang terguling akan dikeluarkan dari arena pertandingan.

- 5.8 Boneka bayi yang tergeser atau terangkat karena kasus 5.1 dan atau 5.2 akan dikembalikan ke tempat semula jika robot pelanggar telah diangkat dari arena pertandingan.

## VI. PENDAFTARAN PESERTA

Tiap Perguruan Tinggi dapat melakukan pendaftaran untuk ikut serta dengan pertamakali mengirimkan proposal ke alamat:

Direktorat Penelitian dan Pengabdian kepada Masyarakat (DP2M)  
Gedung Direktorat Jenderal Pendidikan Tinggi (DIKTI) Lantai 4.  
Departemen Pendidikan Nasional (DEPDIKNAS)  
Jl. Jend. Sudirman, Pintu I Senayan-Jakarta, 10002.

Proposal berisi setidaknya:

- Identitas tim yang terdiri dari satu pembimbing (dosen) dan dua anggota tim (mahasiswa aktif) disertasi dengan lembar pengesahan dari pejabat di perguruan tinggi.
- Bentuk rekaan robot yang akan dibuat disertai penjelasan tentang sistem prosesor, sensor dan aktuator yang akan digunakan.
- Penjelasan secara singkat tentang strategi yang akan digunakan dalam pertandingan.

## VII. BIAYA PEMBUATAN ROBOT, TRANSPORTASI dan AKOMODASI PESERTA

- Untuk pertandingan tingkat regional panitia hanya akan menanggung biaya akomodasi selama berada di lokasi pertandingan. Biaya pembuatan robot dan transportasi adalah tanggungjawab masing-masing tim.
- Untuk pertandingan tingkat nasional, peserta akan mendapat bantuan biaya untuk pembuatan robot, akomodasi dan transportasi dari daerah asal ke venue pertandingan.

## VIII. JADWAL DAN TEMPAT KONTES

Jadwal Lengkap Divisi Pemadam Api Divisi Expert Battle adalah sebagai berikut:

- Pengumuman awal Rule of Game: 01 Desember 2008.
- Pengiriman Proposal: Proposal harus sudah sampai di Gedung DIKTI (lihat VI) paling lambat pada 2 Januari 2009 pk.16:00 WIB.

- c. Pengumuman Tahap I (proposal): mid Januari 2009.
- d. Pengiriman Progress Report berupa video dan slide Power Point tim robot, paling lambat 13 Maret 2008 (alamat sama).
- e. Pengumuman Tahap II (peserta tingkat regional): akhir Maret 2009.
- f. Kontes tingkat regional: disesuaikan dengan jadwal KRI KRCI secara keseluruhan.

#### **IX. PENYELENGGARA**

Direktorat Penelitian dan Pengabdian kepada Masyarakat (DP2M)  
Direktorat Jenderal Pendidikan Tinggi (DIKTI)  
Departemen Pendidikan Nasional (DEPDIKNAS)  
Jl. Jend. Sudirman Pintu I, Senayan-Jakarta, 10002.  
TEL. 021-5700049, 5731251, 5731956 (hunting) ext. 1855.  
FAX. 021-5732468.

#### **X. CONTACT PERSON**

1. **Dr. Ir. Endra Pitowarno, M.Eng** (EEPIS/PENS), Tel: 031-5947280 ext. 2106, HP: 0812.3030.162, Email: [epit@eepis-its.edu](mailto:epit@eepis-its.edu), Mailing List: [krci@groups.eepis-its.edu](mailto:krci@groups.eepis-its.edu), portal <http://kri.eepis-its.edu>

## **BIODATA PENULIS**



Nama : Aloysius Gonzaga Yohanes  
Prasetyo Wijoyo  
NRP : 5103005009  
Tempat / tanggal Lahir : Surabaya, 21 Juni 1984  
Agama : Katolik  
Alamat : Jalan Kebalen Wetan IV/2,  
Surabaya 60163  
E-mail : teknik06@yahoo.com,  
technician06@gmail.com

### **Riwayat Pendidikan**

Sekolah Dasar:

SD Katolik Santa Angela Surabaya, lulusan tahun 1999

Sekolah Menengah Pertama:

SMP Katolik Stella Maris Surabaya, lulusan tahun 2002

Sekolah Menengah Atas:

SMA Katolik Stella Maris Surabaya, lulusan tahun 2005

Universitas :

Sejak tahun 2005 hingga biodata ini ditulis tercatat sebagai mahasiswa  
Fakultas Teknik Jurusan Teknik Elektro, Universitas Katolik Widya  
Mandala Surabaya