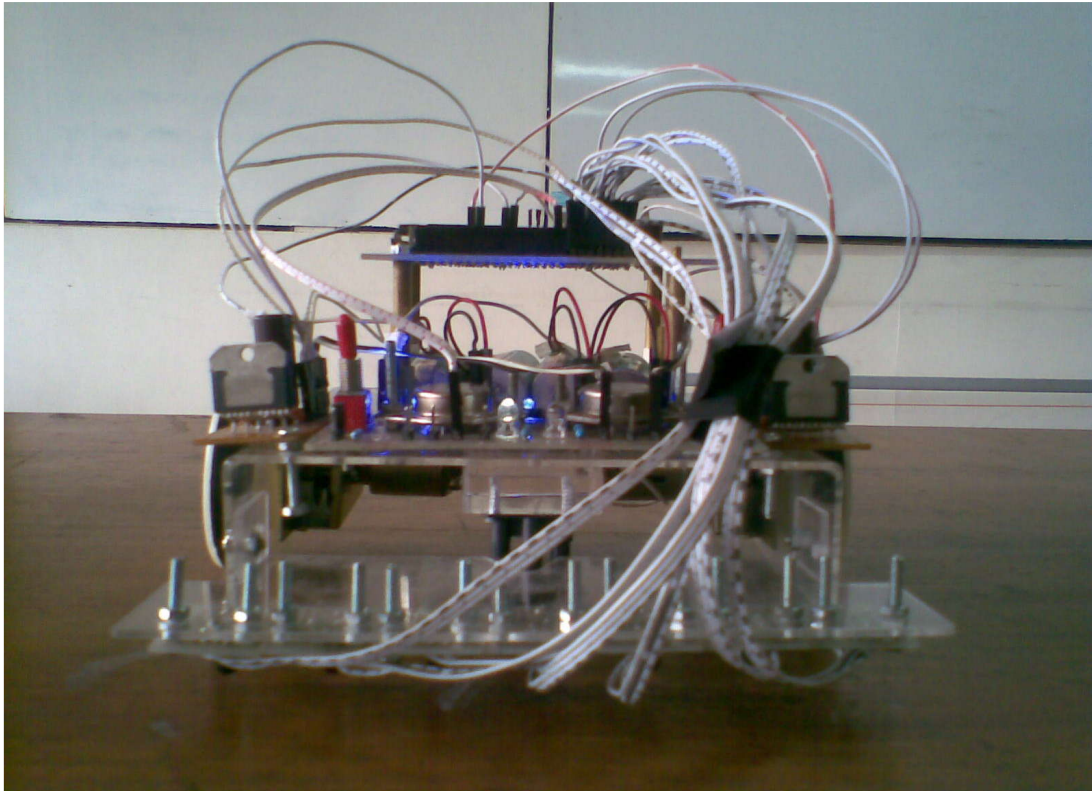
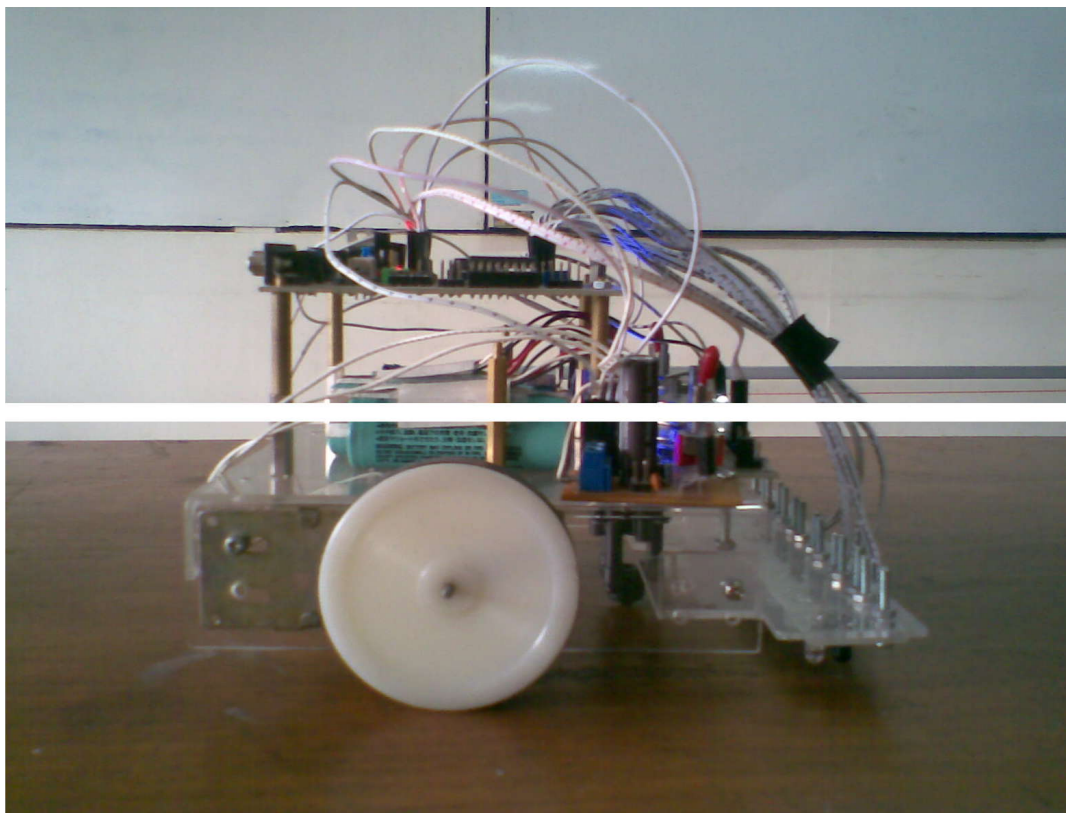


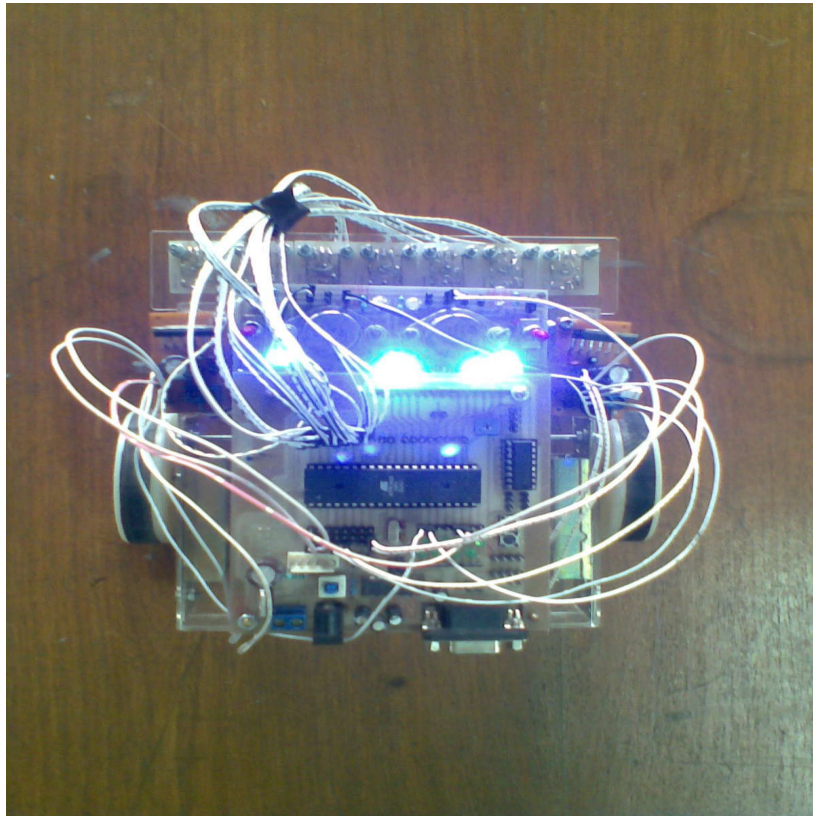
LAMPIRAN A
FOTO ROBOT
&
HARDWARE



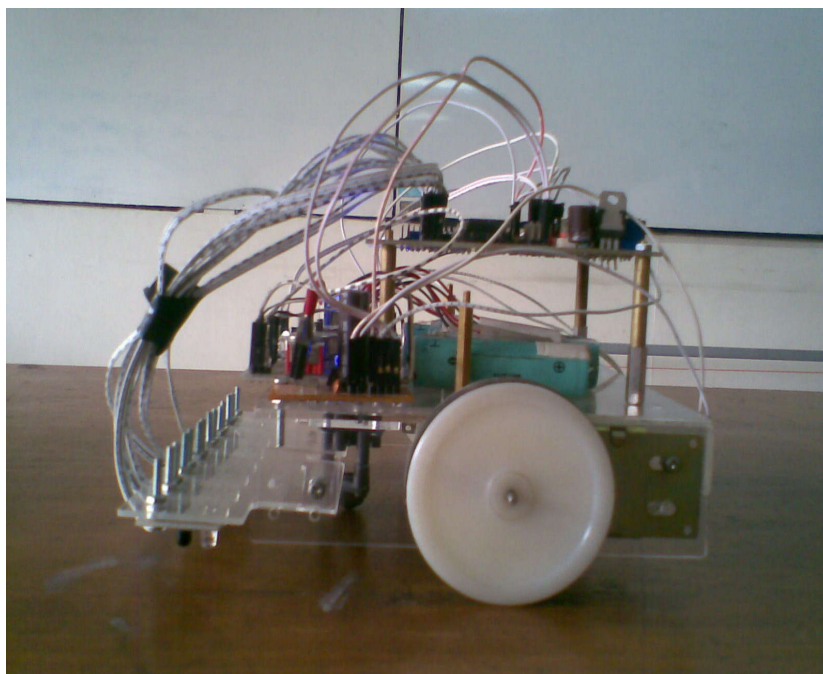
TAMPAK DEPAN – *FRONT VIEW*



TAMPAK SAMPING KANAN – *RIGHT SIDE VIEW*



TAMPAK ATAS – *TOP VIEW*



TAMPAK SAMPING KIRI – *LEFT SIDE VIEW*

LAMPIRAN B

LISTING PROGRAM

```

/*PROGRAM BACA NILAI ADC SENSOR
This program was produced by the
CodeWizardAVR V1.25.3 Standard
Automatic Program Generator
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

```

```

Project :
Version :
Date   : 5/19/2010
Author : F4CG
Company : F4CG
Comments:

```

```

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

```

```

#include <mega16.h>

```

```

// Standard Input/Output functions
#include <stdio.h>
#include <delay.h>

```

```

#define ADC_VREF_TYPE 0x40

```

```

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

```

```

// Declare your global variables here

```

```

void main(void)
{
    int sensor,i;
    // Declare your local variables here

```

```

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

```

```

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In

```

```

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off

```

```

// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

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

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x47;

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

// ADC initialization
// ADC Clock frequency: 172.800 kHz
// ADC Voltage Reference: AVCC pin
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x86;

while (1)
{
    // Place your code here
    for(i=0;i<7;i++)
    {
        sensor=read_adc(i);
        printf("sensor %d=%d",i,sensor);
        putchar(0x0D);//supaya ke bawah

    }
    delay_ms(2000);
};
}

```

```
//Progam Robot pengikut garis menggunakan PID
This program was produced by the
CodeWizardAVR V1.25.9 Standard
Automatic Program Generator
© Copyright 1998-2008 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com
```

```
Project :
Version :
Date   : 6/14/2010
Author : F4CG
Company : F4CG
Comments:
```

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

```
#include <mega16.h>
```

```
#include <delay.h>
```

```
#define ADC_VREF_TYPE 0x40
#define nilai_batas 300
```

```
void putarkiri(void)
{
    OCR1BL=140;    //kec. mtr kanan
    OCR1AL=20;     //kec. mtr kiri
    PORTD.1=1;    //mtr kanan
    PORTD.0=1;    //mtr kiri
}
void putarkanan(void)
{
    OCR1AL=120;    // kec. mtr kiri
    OCR1BL=10;     // kec. mtr kanan
    PORTD.0=1;     // mtr kiri
    PORTD.1=1;     // mtr kanan
}
```

```
int prevSensor, rsensor, lsensor ;
```

```
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
```

```

ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}
int mapSensor(int sensorx)
{
    int s,sensor;
    sensor = sensorx & 0b00111110; // 5 ditengah yang dipake
    switch(sensor)
    {
case 0b00000000:
    s=0;
    break;
case 0b00000010:
    s=10;
    break;
case 0b00000110:
    s=20;
    break;
case 0b00000100:
    s=30;
    break;
case 0b00001110:
    s=40;
    break;
case 0b00001100:
    s=50;
    break;
case 0b00001000: //setpoint
    s=60;
    break;
case 0b00011000:
    s=70;
    break;
case 0b00111000:
    s=80;
    break;
case 0b00010000:
    s=90;
    break;
case 0b00110000:
    s=100;
    break;
case 0b00100000:
    s=110;

```

```

        break;
    }
    return s;
}

```

```

int readline()
{

```

```

    int x, sensor=0, s,i;
    //lsensor = 0;
    //rsensor = 0;
    for(i=0;i<7;i++)
    {
        x=read_adc(i);
        if(x > nilai_batas)
        {
            s=1<< i;
            sensor = sensor|s;

```

```

    }
}

```

```

lsensor = sensor & 0b00000001;//sensor kiri A0
rsensor = sensor & 0b01000000;//sensor kanan A6

```

```

s = mapSensor(sensor & 0b00111110);
if(s > 0)
    prevSensor=s;
else
    s=prevSensor;
return s;
}

```

// Declare your global variables here

```

void main(void)
{
    int maxSpeed=200, setPoint=60;
    int position,motorPower, prevMotorPower=0,err,prevErr=0;
    int p,i,d;

```

// Declare your local variables here

```

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

```

```

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

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out
Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTD=0x00;
DDRD=0xFF;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 10.800 kHz
// Mode: Fast PWM top=00FFh
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0xA1;
TCCR1B=0x0D;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

```

```

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

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

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

// ADC initialization
// ADC Clock frequency: 86.400 kHz
// ADC Voltage Reference: AVCC pin
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x87;
PORTD.0=1;//kiri
PORTD.1=1;//kanan

```

```

while (1)
{
    // Place your code here
    //program untuk garis lintasan lancip dan PID
    position=readline();
    err= setPoint-position;
    p=err*3/2;
    d=(err - prevErr)*3/2;
    i+=err;
    prevErr=err;
    motorPower=p+d+i/1000;
    if(motorPower > maxSpeed)
        motorPower = maxSpeed;
    if(motorPower < - maxSpeed)

```

```

        motorPower = -maxSpeed;
        prevMotorPower = motorPower;
    if(motorPower < 0)
    {
        OCR1BL = maxSpeed + motorPower;
        OCR1AL = maxSpeed;

    }
    else
    {
        OCR1BL = maxSpeed;
        OCR1AL = maxSpeed - motorPower;
    }

    if (rsensor == 1)
    {
        putarkanan();
        delay_ms(700);
    }
    if (lsensor == 1)
    {
        putarkiri();
        delay_ms(800);
    }

};
}

```

LAMPIRAN C

FLOWCHART

Diagram Alir Sistem Keseluruhan

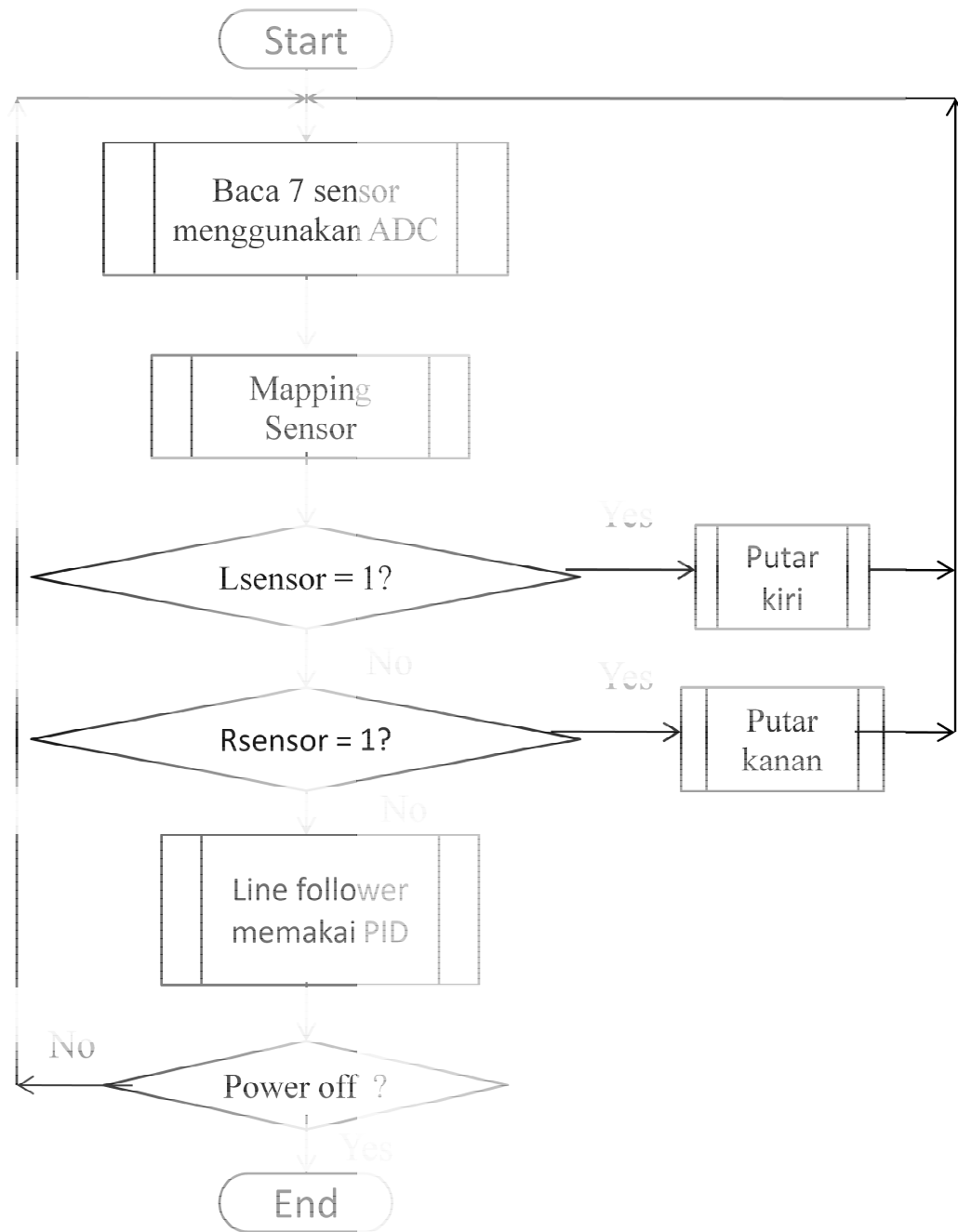


Diagram Alir Ketika Robot Putar Kiri dan Putar Kanan

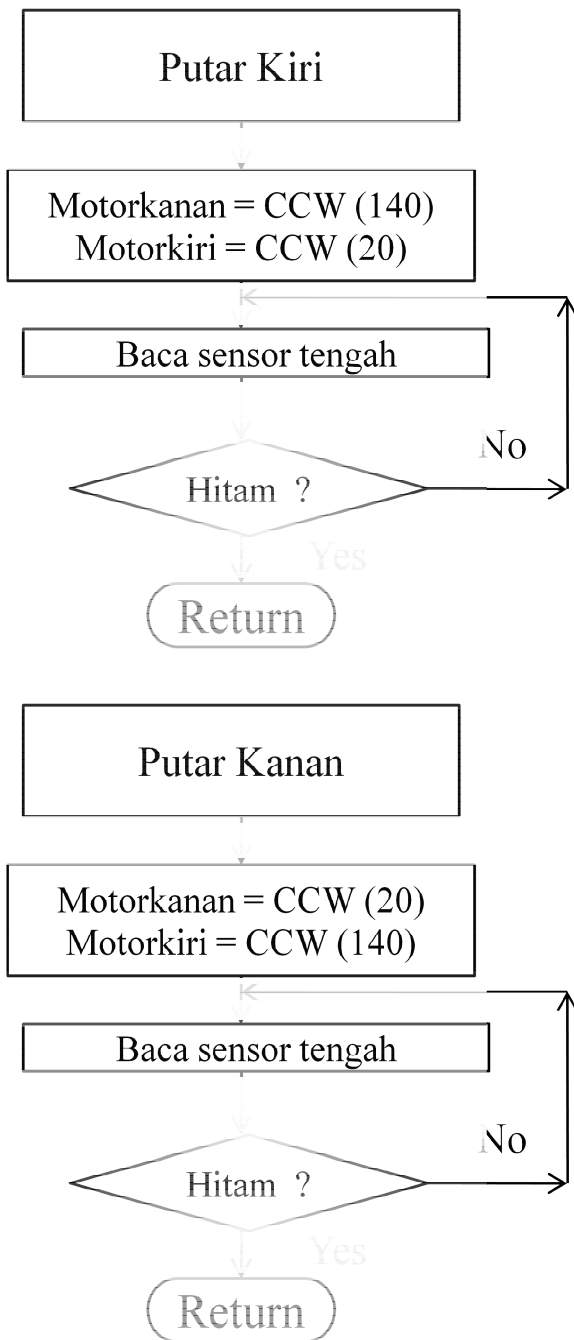


Diagram Alir Pembacaan Nilai ADC 7 Sensor Garis

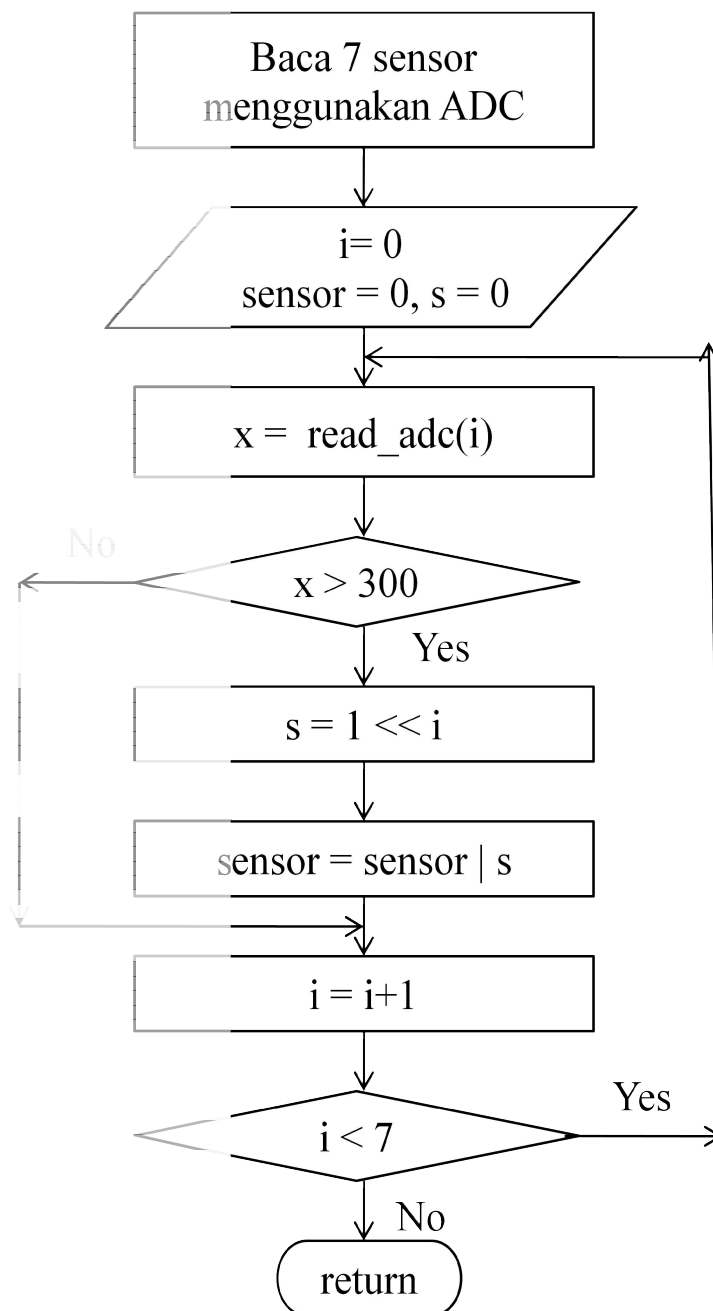
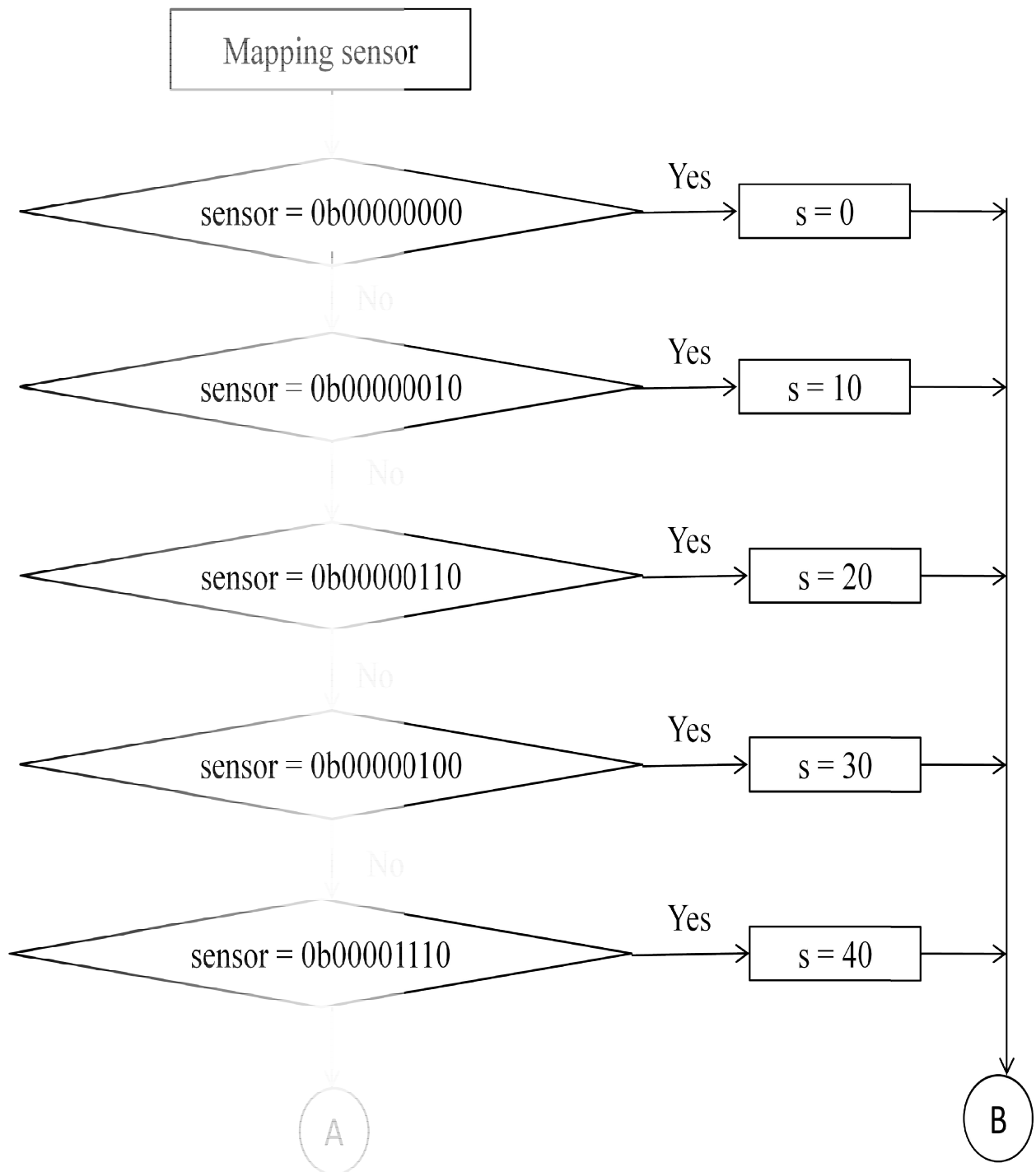


Diagram Alir Pemetaan Sensor



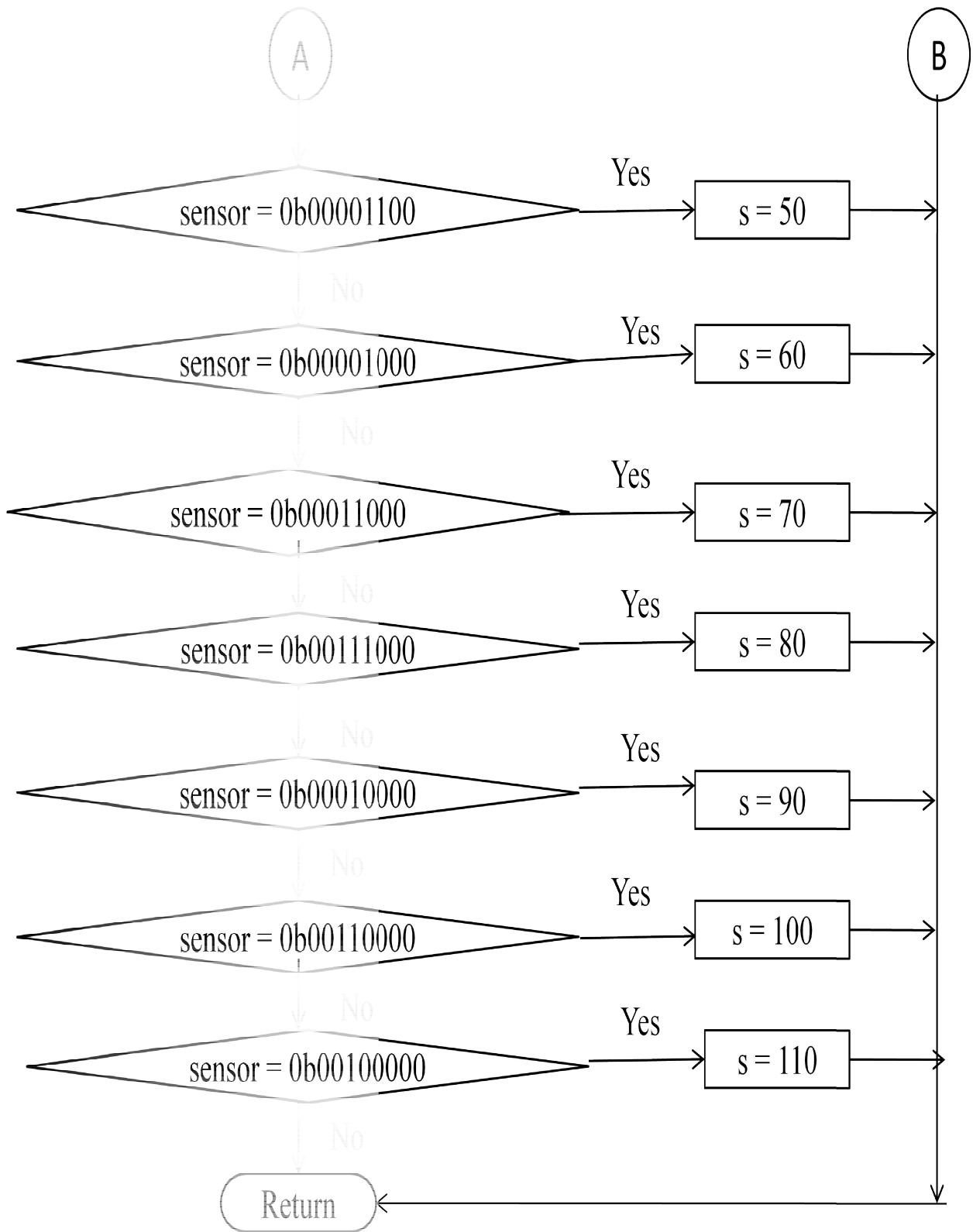
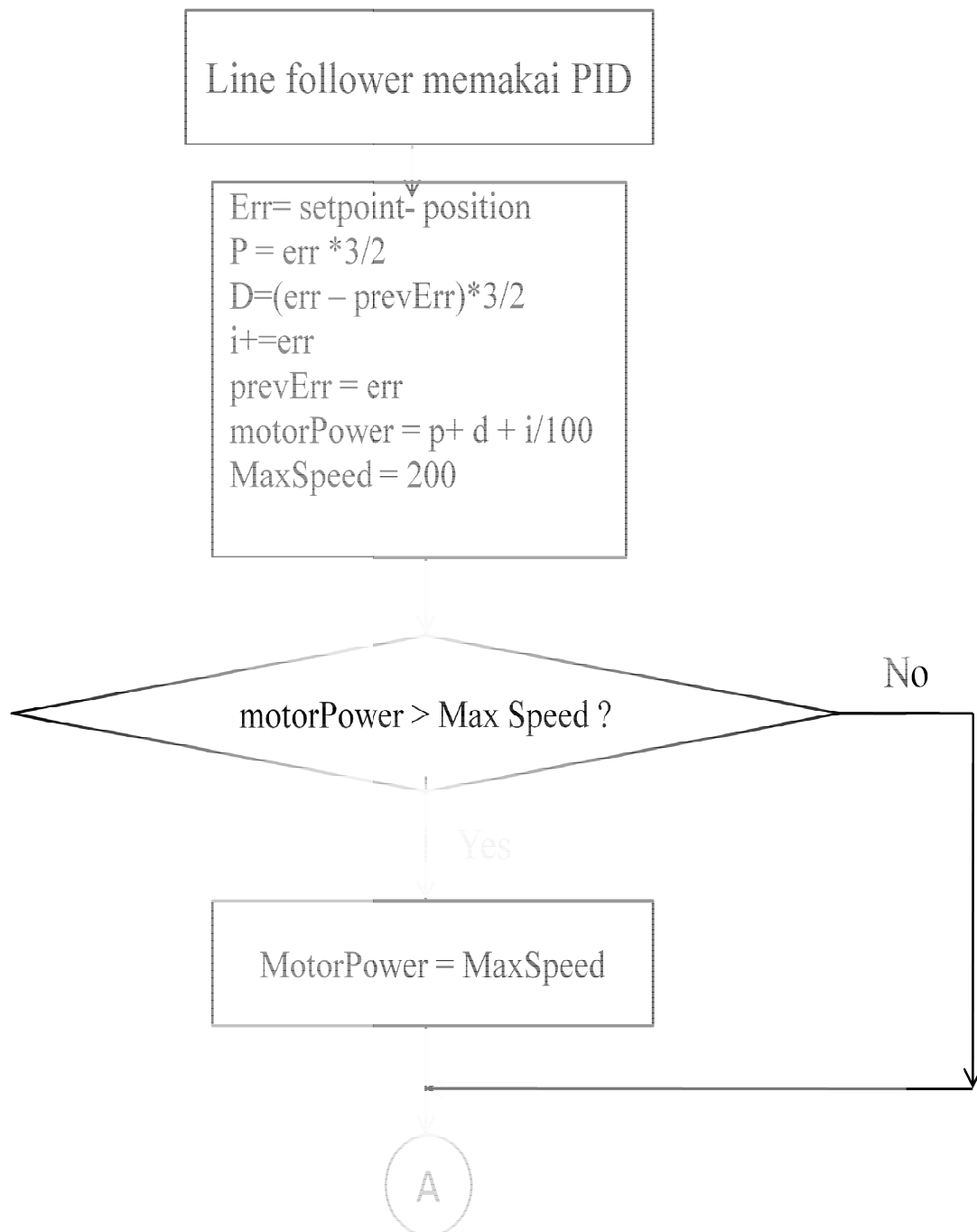


Diagram Alir Robot Pengikut Garis Menggunakan Pengontrol PID



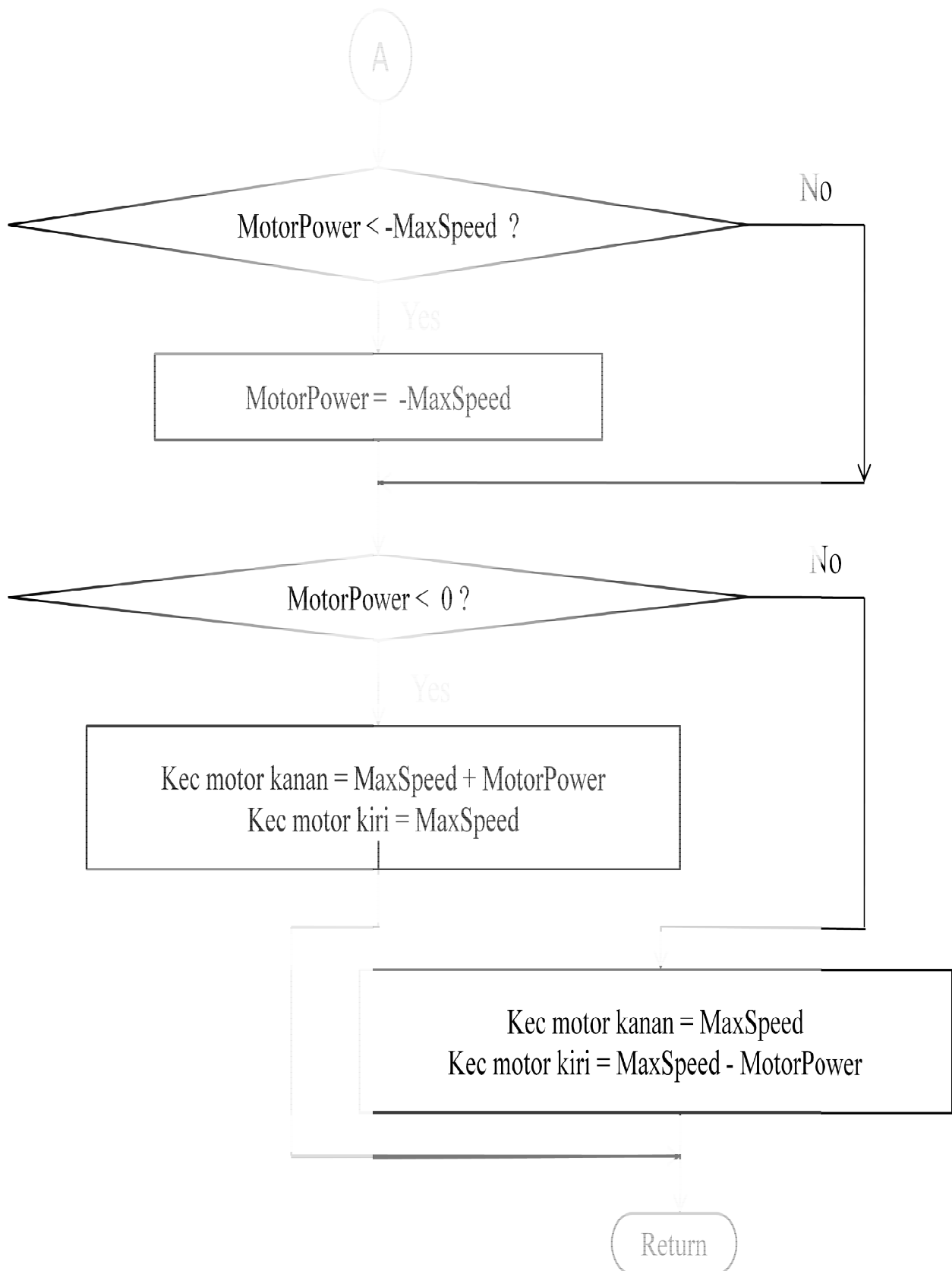


Diagram Alir untuk Mendapatkan Nilai Motor Power (PWM PID)

