

## LAMPIRAN

```
#include <MsTimer2.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(22,24,28,30,32,34);

//-----PIN PUSH BUTTON-----//
#define button1 39
#define button2 41
#define button3 43
#define button4 45
#define button5 47
#define button6 49
#define button7 51
#define button8 53

int PWM, pwm, jarak, durasi, kecepatan;
long pulsa, PULSA;
float const perRotasi = 107;

float JD,JS,JJ;
float KR,KS,KT;
float Rule1, Rule2, Rule3, Rule6, Rule9;
float Rule4, Rule5, Rule7, Rule8;
float R1,R2,R3,R4,R5,R6,R7,R8,R9;
float PS1, PS2, PS3,PS4,PS5,PS6,PS7;
float P1, P2;
float Perbandingan_min, Perbandingan_max;
float Defuzifikasi, Presentase, Hasil_Presentase, Pengereman;

//-----PIN ULTRASONIK-----//
#define TRIGPIN 8
#define ECHOPIN 9

//-----PIN MOTOR DC-----//
#define enA 7
#define in1 6
#define in2 5
```

```
#define enC1 21
#define enD1 36

void setup() {
    // put your setup code here, to run once:
    Serial.begin (9600);
    lcd.begin(16,4);

    pinMode (button1,INPUT);
    pinMode (button2, INPUT);
    pinMode (button3, INPUT);
    pinMode (button4, INPUT);
    pinMode (button5, INPUT);
    pinMode (button6, INPUT);
    pinMode (button7, INPUT);
    pinMode (button8, INPUT);

    pinMode (TRIGPIN,OUTPUT);
    pinMode (ECHOPIN,INPUT);

    MsTimer2::set (100,interrupt0);
    MsTimer2::start();

    attachInterrupt (digitalPinToInterrupt (enC1), interrupt,CHANGE);

    pinMode (enA, OUTPUT);
    pinMode (in1, OUTPUT);
    pinMode (in2, OUTPUT);
    pinMode (enC1, INPUT_PULLUP);
    pinMode (enD1, INPUT_PULLUP);

    digitalWrite (in1,LOW);
    digitalWrite (in2,HIGH);
    int PWM;
}

void loop() {
    // put your main code here, to run repeatedly:
```

```
pb();
sensor();
inisialisasi();
tampil_sensor();
tampil_kecepatan();
tampil_fuzzy();
}

void pb() {
    int but1 = digitalRead(button1);
    int but2 = digitalRead(button2);
    int but3 = digitalRead(button3);
    int but4 = digitalRead(button4);
    int but5 = digitalRead(button5);
    int but6 = digitalRead(button6);
    int but7 = digitalRead(button7);
    int but8 = digitalRead(button8);

    if (but1 == HIGH) {
        PWM = 0;
        analogWrite(enA, PWM);
    }

    if (but2 == HIGH) {
        PWM = 50;
        analogWrite(enA, PWM);
    }

    if (but3 == HIGH) {
        PWM = 100;
        analogWrite(enA, PWM);
    }

    if (but4 == HIGH) {
        PWM = 150;
        analogWrite(enA, PWM);
    }
}
```

```

if (but5 == HIGH) {
    PWM = 200;
    analogWrite(enA, PWM);
}

if (but6 == HIGH) {
    PWM = 255;
    analogWrite(enA, PWM);
}

Hasil_Presentase = ( (DF2(A2)*0.01) * PWM);
Pengereman = ( PWM - Hasil_Presentase );

Serial.print("Hasil_Presentase = ");
Serial.println(Hasil_Presentase);

Serial.print("Pengereman = ");
Serial.print(Pengereman);

if (Pengereman > 0)
    return analogWrite(enA,Pengereman);
else {
    return analogWrite (enA, 0);
}
}

void inisialisasi() {
    JD=jarak_dekat (jarak);
    JS=jarak_sedang (jarak);
    JJ=jarak_jauh (jarak);
    KR=kecepatan_rendah (kecepatan);
    KS=kecepatan_sedang (kecepatan);
    KT=kecepatan_tinggi (kecepatan);
    Rule1 = RB1 (R1);
    Rule2 = RB2 (R2);
    Rule3 = RB3 (R3);
    Rule4 = RB4 (R4);
    Rule5 = RB5 (R5);
}

```

```

Rule6 = RB6 (R6);
Rule7 = RB7 (R7);
Rule8 = RB8 (R8);
Rule9 = RB9 (R9);
Perbandingan_min = DEF1 (P1);
Perbandingan_max = DEF2 (P2);
Defuzifikasi = DF1 (A1);
Presentase = DF2 (A2);
}

void sensor() {
    digitalWrite(TRIGPIN,LOW);
    delayMicroseconds(10);

    digitalWrite(TRIGPIN,HIGH);
    delayMicroseconds(20);
    digitalWrite(TRIGPIN,LOW);

    durasi = pulseIn(ECHOPIN,HIGH);
    jarak = durasi*0.034/2;
}

void tampil_sensor() {
    lcd.setCursor(0,0);
    lcd.print("                ");
    lcd.setCursor(0,0);
    lcd.print("Jarak = ");
    lcd.print(jarak);
    lcd.print(" cm");
    delay(1000);
}

void interrupt0() {
    kecepatan = pulsa*60*0.65/perRotasi;
    pulsa=0;
}

void interrupt() {

```

```

if(digitalRead(enC1)!=digitalRead(enD1)){
    pulsa++;
}
else{
    pulsa--;
}
PULSA=pulsa/3;
}

void tampil_kecepatan() {
    lcd.setCursor(0,1);
    lcd.print("          ");
    lcd.setCursor(0,1);
    lcd.print("Kec = ");
    lcd.print(kecepatan);
    lcd.print(" k/j");
    delay(1000);
}

void tampil_fuzzy() {
    //-----TAMPILAN DERAJAT JARAK-----//
    Serial.print("DKJD = ");
    Serial.println(jarak_dekat(jarak));

    Serial.print("DKJS = ");
    Serial.println(jarak_sedang(jarak));

    Serial.print("DKJJ = ");
    Serial.println(jarak_jauh(jarak));

    //-----TAMPILAN DERAJAT KECEPATAN-----//
    Serial.print("DKKR = ");
    Serial.println(kecepatan_rendah(kecepatan));

    Serial.print("DKKS = ");
    Serial.println(kecepatan_sedang(kecepatan));

    Serial.print("DKKT = ");

```

```
Serial.println(kecepatan_tinggi(kecepatan));

//-----TAMPILAN RULE BASED-----//
Serial.print("Aturan ke 1 = ");
Serial.println(RB1(R1));

Serial.print("Aturan ke 2 = ");
Serial.println(RB2(R2));

Serial.print("Aturan ke 3 = ");
Serial.println(RB3(R3));

Serial.print("Aturan ke 4 = ");
Serial.println(RB4(R4));

Serial.print("Aturan ke 5 = ");
Serial.println(RB5(R5));

Serial.print("Aturan ke 6 = ");
Serial.println(RB6(R6));

Serial.print("Aturan ke 7 = ");
Serial.println(RB7(R7));

Serial.print("Aturan ke 8 = ");
Serial.println(RB8(R8));

Serial.print("Aturan ke 9 = ");
Serial.println(RB9(R9));

//-----TAMPILAN MIN MAX PENGEREMAN-----//
Serial.print("PS1 = ");
Serial.println(max1(PS1));

Serial.print("PS2 = ");
Serial.println(max2(PS2));

Serial.print("PS3 = ");
```

```

Serial.println(max3(PS3));

Serial.print("PS4 = ");
Serial.println(max4(PS4));

Serial.print("PS5 = ");
Serial.println(max5(PS5));

Serial.print("PS6 = ");
Serial.println(max6(PS6));

Serial.print("PS7 = ");
Serial.println(max7(PS7));

//-----TAMPILAN PERBANDINGAN MIN MAX-----//
Serial.print("DEF 1 = ");
Serial.println(DEF1(P1));

Serial.print("DEF 2 = ");
Serial.println(DEF2(P2));

//-----TAMPILAN DEFUZIFIKASI-----//
Serial.print("Defuzifikasi = ");
Serial.println(DF1 (A1));

Serial.print("Presentase = ");
Serial.println(DF2 (A2));
}

//-----FUZZIFIKASI-----//
//---JARAK---//
float jarak_dekat (float jarak){
    if (jarak<=3)
        return(1);
    else{
        if (jarak<=10)
            return((10-jarak)/7);
        else{

```



```
        return(0);
    }
}

float jarak_sedang (float jarak){
    if (jarak <=8)
        return(0);
    else{
        if (jarak<=17)
            return((jarak-8)/9);
        else{
            if (jarak<=30)
                return((30-jarak)/13);
            else{
                return(0);
            }
        }
    }
}

float jarak_jauh (float jarak){
    if (jarak<=25)
        return(0);
    else{
        if (jarak<=35)
            return((jarak-25)/10);
        else{
            if (jarak<=100)
                return(1);
            else{
                return(0);
            }
        }
    }
}

//---KECEPATAN---//
```

```
float kecepatan_rendah (float kecepatan){
    if (kecepatan<=12)
        return(1);
    else{
        if (kecepatan<=50)
            return((50-kecepatan)/38);
        else{
            return(0);
        }
    }
}
```

```
float kecepatan_sedang (float kecepatan){
    if (kecepatan <=40)
        return(0);
    else{
        if (kecepatan<=60)
            return((kecepatan-40)/20);
        else{
            if (kecepatan<=90)
                return((90-kecepatan)/30);
            else{
                return(0);
            }
        }
    }
}
```

```
float kecepatan_tinggi (float kecepatan){
    if (kecepatan<=80)
        return(0);
    else{
        if (kecepatan<=100)
            return((kecepatan-80)/20);
        else{
            return(0);
        }
    }
}
```

```
    }

//-----RULE BASED-----//
//---MIN RULE BASED---//
float RB1 (float R1)
{
    if (KR < JD)
        return R1 = KR;
    else
    {
        return R1 = JD;
    }
}

float RB2 (float R2)
{
    if (KR < JS)
        return R2 = KR;
    else
    {
        return R2 = JS;
    }
}

float RB3 (float R3)
{
    if (KR < JJ)
        return R3 = KR;
    else
    {
        return R3 = JJ;
    }
}

float RB4 (float R4)
{
    if (KS < JD)
        return R4 = KS;
```

```
    else
    {
        return R4 = JD;
    }
}

float RB5 (float R5)
{
    if (KS < JS)
        return R5 = KS;
    else
    {
        return R5 = JS;
    }
}

float RB6 (float R6)
{
    if (KS < JJ)
        return R6 = KS;
    else
    {
        return R6 = JJ;
    }
}

float RB7 (float R7)
{
    if (KT < JD)
        return R7 = KT;
    else
    {
        return R7 = JD;
    }
}

float RB8 (float R8)
{
```

```
    if (KT < JS)
    return R8 = KT;
    else
    {
        return R8 = JS;
    }
}

float RB9 (float R9)
{
    if (KT < JJ)
    return R9 = KT;
    else
    {
        return R9 = JJ;
    }
}

//---MAX RULE BASED---//
//PENGEREMAN LEMAH//
float max1 (float PS1)
{
    if (RB1(R1) < RB2(R2))
    return PS1 = RB2(R2);
    else
    {
        return PS1 = RB1(R1);
    }
}

float max2 (float PS2)
{
    if (max1(PS1) < RB3(R3))
    return PS2 = RB3(R3);
    else
    {
        return PS2 = max1(PS1);
    }
}
```

```
}

float max3 (float PS3)
{
  if (max2(PS2) < RB6(R6))
    return PS3 = RB6(R6);
  else
  {
    return PS3 = max2(PS2);
  }
}

float max4 (float PS4)
{
  if (max3(PS3) < RB9(R9))
    return PS4 = RB9(R9);
  else
  {
    return PS4 = max3(PS3);
  }
}

//PENGGEREMAN KUAT//
float max5 (float PS5)
{
  if (RB4(R4) < RB5(R5))
    return PS5 = RB5(R5);
  else
  {
    return PS5 = RB4(R4);
  }
}

float max6 (float PS6)
{
  if (max5(R5) < RB7(R7))
    return PS6 = RB7(R7);
  else
```

```

    {
        return PS6 = max5(PS5);
    }
}

float max7 (float PS7)
{
    if (max6(PS6) < RB8(R8))
        return PS7 = RB8(R8);
    else
    {
        return PS7 = max6(PS6);
    }
}

//---DERAJAT PENGGEREMAN LEMAH---//
float DEF1 (float P1) {
    if (max1(PS1) > max2(PS2) && max1(PS1) > max3(PS3) && max1(PS1)
> max4(PS4))
        return P1 = max1(PS1);

    else if (max2(PS2) > max1(PS1) && max2(PS2) > max3(PS3) &&
max2(PS2) > max4(PS4))
        return P1 = max2(PS2);

    else if (max3(PS3) > max1(PS1) && max3(PS3) > max2(PS2) &&
max3(PS3) > max4(PS4))
        return P1 = max3(PS3);

    else {
        return P1 = max4(PS4);
    }
}

//---DERAJAT PENGGEREMAN KUAT---//
float DEF2 (float P2)
{
    if (max5(PS5) > max6(PS6) && max5(PS5) > max7(PS7))

```

```
return P2 = max5(PS5);

else if (max6(PS6) > max5(PS5) && max6(PS6) > max7(PS7))
return P2 = max6(PS6);

else {
    return P2 = max7(PS7);
}

//-----DEFUZIFIKASI-----//
float DF1 (float A1)
{
    return Defuzifikasi = ( (DEF1(P1)*1)+(DEF2(P2)*5) ) /
(DEF1(P1)+DEF2(P2) );
}

float DF2 (float A2)
{
    return Presentase = ( DF1(A1)*2*10 );
}
```