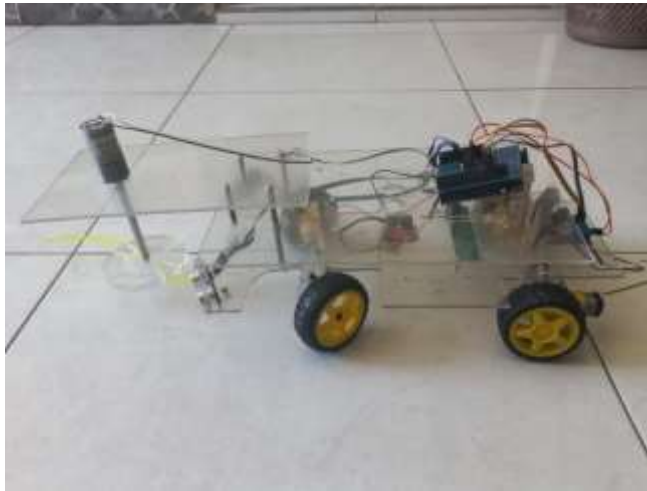


LAMPIRAN



Code

```
/*
  FUZZY LOGIC
*/

float inputKe1 = 44;
float inputKe2 = 27;
#include <L298N.h>
const int ENA = 6; // motor A (kiri)
const int IN1 = 8;
const int IN2 = 7;
const int IN3 = 2;
const int IN4 = 4;
const int ENB = 3; // motor B (kanan)
L298N driver(ENA, IN1, IN2, IN3, IN4, ENB);
#define TRIGGER_kanan 10 // Pin trigger sensor
ultrasonik terhubung ke pin 2 Arduino
#define ECHO_kanan 9 // Pin echo sensor ultrasonik
terhubung ke pin 3 Arduino
#define TRIGGER_kiri 12 // Pin trigger sensor
ultrasonik terhubung ke pin 2 Arduino
#define ECHO_kiri 11 // Pin echo sensor ultrasonik
terhubung ke pin 3 Arduino
char *GARIS[] = {"NOL", "NAIK", "FLAT", "TURUN"};
```

```

char *OUTPUTUT[] = {"LURUS", "SERONG KANAN", "KANAN
TAJAM", "SERONG KIRI", "KIRI TAJAM"};

float MFoutput[] = {90,60,30,120, 150};

// membership function SENSOR ke-1
float MF1 [3][4] = { { 0, 0, 12, 22},
                    {12, 22, 35, 45},
                    {35, 45, 60, 60}};

char *MF1_kondisi[] = {"DEKAT", "SEDANG", "JAUH"};
char *MF1_garis[] = {"", "", ""};
float MF1_nilai[3];

// membership function SENSOR ke-2
float MF2 [3][4] = { { 0, 0, 12, 22},
                    {12, 22, 35, 45},
                    {35, 45, 60, 60}};

char *MF2_kondisi[] = {"DEKAT", "SEDANG", "JAUH"};
char *MF2_garis[] = {"", "", ""};
float MF2_nilai[3];

// RULES
char *urutanRULES[9] = {"KANAN TAJAM", "KIRI TAJAM",
"KIRI TAJAM",
"KANAN TAJAM", "SERONG KANAN",
"SERONG KIRI",
"KANAN TAJAM", "SERONG
KIRI", "LURUS"};
float RULESdetail [9][4];
float RULEScariMAX [9][5];
float RULESMAX [5];

float COG_data [4][3];
float COG_sum1=0;
float COG_sum2=0;
float FINAL;

float terbesar=0;
int speed = 125;

```

```

void ukur_Jarak(){
    int duration, duration2;
    // Mengirimkan pulsa ultrasonik selama 10 mikrodetik
    digitalWrite(TRIGGER_kanan, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIGGER_kanan, HIGH);
    delayMicroseconds(8);
    digitalWrite(TRIGGER_kanan, LOW);
    digitalWrite(TRIGGER_kiri, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIGGER_kiri, HIGH);
    delayMicroseconds(8);
    digitalWrite(TRIGGER_kiri, LOW);

    // Membaca durasi pulsa yang dikembalikan oleh sensor
    ultrasonik
    duration = pulseIn(ECHO_kanan, HIGH);
    duration2 = pulseIn(ECHO_kiri, HIGH);

    // Menghitung jarak berdasarkan durasi
    inputKe1 = duration * 0.034 / 2;
    inputKe2 = duration2 * 0.034 / 2;
    Serial.print("1 ");
    Serial.println(inputKe1);
}
void tabelRules_MAX(){

    int i=0,j=0;
    // cari data di tabel CARI MAX
    for (i = 0; i < 9; i++) {
        RULEScariMAX[i][0]=0;
        RULEScariMAX[i][1]=0;
        RULEScariMAX[i][2]=0;
        RULEScariMAX[i][3]=0;
        RULEScariMAX[i][4]=0;
        // kolom 1 (INDEX nya 0)
        if(RULESdetail[i][3]==1)
RULEScariMAX[i][0]=RULESdetail[i][2]; // kolom MATI,
jadi harus 1
        // kolom 2 (INDEX nya 1)
        if(RULESdetail[i][3]==2)

```

```

RULEScariMAX[i][1]=RULESdetail[i][2]; // kolom PELAN,
jadi harus 2
    // kolom 3 (INDEX nya 2)
    if(RULESdetail[i][3]==3)
RULEScariMAX[i][2]=RULESdetail[i][2]; // kolom SEDANG,
jadi harus 3
    // kolom 4 (INDEX nya 3)
    if(RULESdetail[i][3]==4)
RULEScariMAX[i][3]=RULESdetail[i][2]; // kolom CEPAT,
jadi harus 4
    if(RULESdetail[i][3]==5)
RULEScariMAX[i][4]=RULESdetail[i][2]; // kolom CEPAT,
jadi harus 4

}
// cari data MAX dari tabel diatas
Serial.println("#####");
terbesar=0.0;
for (i = 0; i < 9; i++) {
    if(RULEScariMAX[i][0]>terbesar)
terbesar=RULEScariMAX[i][0];
    //Serial.print(RULEScariMAX[i][0]);
Serial.print("\t");
}
RULESMAX[0]=terbesar;
//
terbesar=0.0;
for (i = 0; i < 9; i++) {
    if(RULEScariMAX[i][1]>terbesar)
terbesar=RULEScariMAX[i][1];
    //Serial.print(RULEScariMAX[i][1]);
Serial.print("\t");
}
RULESMAX[1]=terbesar;
//
terbesar=0.0;
for (i = 0; i < 9; i++) {
    if(RULEScariMAX[i][2]>terbesar)
terbesar=RULEScariMAX[i][2];
    //Serial.print(RULEScariMAX[i][2]);
Serial.print("\t");

```

```

}
RULESMAX[2]=terbesar;
//
terbesar=0.0;
for (i = 0; i < 9; i++) {
    if(RULEScariMAX[i][3]>terbesar)
terbesar=RULEScariMAX[i][3];
    //Serial.print(RULEScariMAX[i][3]);
Serial.print("\t");
}
RULESMAX[3]=terbesar;

terbesar=0.0;
for (i = 0; i < 9; i++) {
    if(RULEScariMAX[i][4]>terbesar)
terbesar=RULEScariMAX[i][4];
    //Serial.print(RULEScariMAX[i][3]);
Serial.print("\t");
}
RULESMAX[4]=terbesar;
}
void calculate_COG(){
    int i=0,j=0;
    //hitung tabel data nya COD
    for (i = 0; i < 5; i++) {
        COG_data[i][0] = RULESMAX[i]; // nilai MAX dari
RULESMAX
        COG_data[i][1] = MFoutput[i]; // nilai dari MF
output
        COG_data[i][2] = COG_data[i][0]*COG_data[i][1]; //
hasil kali nya
    }

    COG_sum1=0;
    COG_sum2=0;
    for (i = 0; i < 5; i++) {
        COG_sum1 += COG_data[i][0]; // cari SUM nya MAX
        COG_sum2 += COG_data[i][2]; // cari SUM nya hasil
perkalian
    }

```

```

FINAL = COG_sum2 / COG_sum1;

}
void tabelRulesDetail(){
    int i=0,j=0;
    for (i = 0; i < 9; i++) {
        // kolom 1
        if(i<3){RULESdetail[i][0]=MF1_nilai[0];}
        else if(i<6){RULESdetail[i][0]=MF1_nilai[1];}
        else {RULESdetail[i][0]=MF1_nilai[2];}
        // kolom 2
        RULESdetail[i][1]=MF2_nilai[j];
        j++; if(j==3)j=0;
        // kolom 3 cari nilai MIN antara kolom 1 dan kolom
2
        if(RULESdetail[i][0]>RULESdetail[i][1])
RULESdetail[i][2]=RULESdetail[i][1];
        else RULESdetail[i][2]=RULESdetail[i][0];
        // kolom 4 kondisinya 1=MATI 2=PELAN 3=SEDANG
4=CEPAT
        if(RULESdetail[i][2]>0){
            if(urutanRULES[i]=="MATI") RULESdetail[i][3] = 1;
            else if(urutanRULES[i]=="SEBENTAR")
RULESdetail[i][3] = 2;
            else if(urutanRULES[i]=="SEDANG")
RULESdetail[i][3] = 3;
            else RULESdetail[i][3] = 4;
        }
    }
}

void cetak_MF1dan2(){
    Serial.print(MF1_garis[0]);Serial.print(",
");Serial.print(MF1_garis[1]);Serial.print(",
");Serial.print(MF1_garis[2]); Serial.print(" === ");
    Serial.print(MF1_nilai[0]);Serial.print(",
");Serial.print(MF1_nilai[1]);Serial.print(",
");Serial.println(MF1_nilai[2]);

    Serial.print(MF2_garis[0]);Serial.print(",

```

```

");Serial.print(MF2_garis[1]);Serial.print(",
");Serial.print(MF2_garis[2]);Serial.print(", ");
Serial.print(" === ");
    Serial.print(MF2_nilai[0]);Serial.print(",
");Serial.print(MF2_nilai[1]);Serial.print(",
");Serial.print(MF2_nilai[2]);Serial.print(",
");Serial.println();
}
void cariMF(){
    // SENSOR ke-1
    if(inputKe1<=MF1[0][0])
{MF1_garis[0]=GARIS[0];MF1_nilai[0]=0;}
    else if (inputKe1<MF1[0][1])
{MF1_garis[0]=GARIS[1];MF1_nilai[0]=(inputKe1-
MF1[0][0])/(MF1[0][1]-MF1[0][0]);}
    else if (inputKe1<=MF1[0][2])
{MF1_garis[0]=GARIS[2];MF1_nilai[0]=1;}
    else if (inputKe1<MF1[0][3])
{MF1_garis[0]=GARIS[3];MF1_nilai[0]=(MF1[0][3]-
inputKe1)/(MF1[0][3]-MF1[0][2]);}
    else {MF1_garis[0]=GARIS[0];MF1_nilai[0]=0;}

    if(inputKe1<=MF1[1][0])
{MF1_garis[1]=GARIS[0];MF1_nilai[1]=0;}
    else if (inputKe1<MF1[1][1])
{MF1_garis[1]=GARIS[1];MF1_nilai[1]=(inputKe1-
MF1[1][0])/(MF1[1][1]-MF1[1][0]);}
    else if (inputKe1<=MF1[1][2])
{MF1_garis[1]=GARIS[2];MF1_nilai[1]=1;}
    else if (inputKe1<MF1[1][3])
{MF1_garis[1]=GARIS[3];MF1_nilai[1]=(MF1[1][3]-
inputKe1)/(MF1[1][3]-MF1[1][2]);}
    else {MF1_garis[1]=GARIS[0];MF1_nilai[1]=0;}

    if(inputKe1<=MF1[2][0])
{MF1_garis[2]=GARIS[0];MF1_nilai[2]=0;}
    else if (inputKe1<MF1[2][1])
{MF1_garis[2]=GARIS[1];MF1_nilai[2]=(inputKe1-
MF1[2][0])/(MF1[2][1]-MF1[2][0]);}
    else if (inputKe1<=MF1[2][2])

```

```

{MF1_garis[2]=GARIS[2];MF1_nilai[2]=1;}
  else if (inputKe1<MF1[2][3])
{MF1_garis[2]=GARIS[3];MF1_nilai[2]=(MF1[2][3]-
inputKe1)/(MF1[2][3]-MF1[2][2]);}
  else {MF1_garis[2]=GARIS[0];MF1_nilai[2]=0;}

// SENSOR ke-2
// dingin
if(inputKe2<=MF2[0][0])
{MF2_garis[0]=GARIS[0];MF2_nilai[0]=0;}
  else if (inputKe2<MF2[0][1])
{MF2_garis[0]=GARIS[1];MF2_nilai[0]=(inputKe2-
MF2[0][0])/(MF2[0][1]-MF2[0][0]);}
  else if (inputKe2<=MF2[0][2])
{MF2_garis[0]=GARIS[2];MF2_nilai[0]=1;} // benar
  else if (inputKe2<MF2[0][3])
{MF2_garis[0]=GARIS[3];MF2_nilai[0]=(MF2[0][3]-
inputKe2)/(MF2[0][3]-MF2[0][2]);}
  else {MF2_garis[0]=GARIS[0];MF2_nilai[0]=0;}
// sejuk
if(inputKe2<=MF2[1][0])
{MF2_garis[1]=GARIS[0];MF2_nilai[1]=0;}
  else if (inputKe2<MF2[1][1])
{MF2_garis[1]=GARIS[1];MF2_nilai[1]=(inputKe2-
MF2[1][0])/(MF2[1][1]-MF2[1][0]);}
  else if (inputKe2<=MF2[1][2])
{MF2_garis[1]=GARIS[2];MF2_nilai[1]=1;}
  else if (inputKe2<MF2[1][3])
{MF2_garis[1]=GARIS[3];MF2_nilai[1]=(MF2[1][3]-
inputKe2)/(MF2[1][3]-MF2[1][2]);}
  else {MF2_garis[1]=GARIS[0];MF2_nilai[1]=0;}
// normal
if(inputKe2<=MF2[2][0])
{MF2_garis[2]=GARIS[0];MF2_nilai[2]=0;}
  else if (inputKe2<MF2[2][1])
{MF2_garis[2]=GARIS[1];MF2_nilai[2]=(inputKe2-
MF2[2][0])/(MF2[2][1]-MF2[2][0]);}
  else if (inputKe2<=MF2[2][2])
{MF2_garis[2]=GARIS[2];MF2_nilai[2]=1;}
  else if (inputKe2<MF2[2][3])
{MF2_garis[2]=GARIS[3];MF2_nilai[2]=(MF2[2][3]-

```



```

inputKe2)/(MF2[2][3]-MF2[2][2]);}
  else {MF2_garis[2]=GARIS[0];MF2_nilai[2]=0;}
}

void setup()
{
  int speed= 125;
  int i=0,j=0;

  pinMode(13, OUTPUT);
  pinMode(9, OUTPUT);
  Serial.begin(9600);
  Serial.println(inputKe1);
  Serial.println(inputKe2);
  ukur_Jarak();
  cariMF(); // hitung

  tabelRulesDetail(); // hitung

  tabelRules_MAX();
  calculate_COG();

  Serial.println("-----cetak MF-----
  -----");
  cetak_MF1dan2();
  Serial.println("-----cetak data tabel rules---
  -----");
  for (int i = 0; i < 9; i++) {
    Serial.print("ke-");Serial.print(i+1);
  Serial.print("\t");
    for (int j = 0; j < 4; j++) {
      Serial.print(RULESdetail[i][j]);
  Serial.print("\t");
    }
    Serial.println("");
  }
  Serial.println("-----cetak data tabel rules
  MAX-----");
  for (int i = 0; i < 9; i++) {
    Serial.print("ke-");Serial.print(i+1);

```

```

Serial.print("\t");
  for (int j = 0; j < 5; j++) {
    Serial.print(RULEScariMAX[i][j]);
Serial.print("\t");
  }
  Serial.println("");
}
Serial.println("-----rules max-----
-----");

Serial.print("hasil rules MAX = ");
for (i = 0; i < 5; i++) {
  Serial.print(RULESMAX[i]); Serial.print("\t");

}

Serial.println("");
Serial.println("-----data tabel COG-----
-----");
//COG_data

for (i = 0; i < 5; i++) {
  for (j = 0; j < 3; j++) {
    Serial.print(COG_data[i][j]); Serial.print("\t");
  }
  Serial.println("");
}
//-----
Serial.print("hasil FINAL = ");
Serial.println(FINAL);
Serial.println("-----");

}

void loop()
{
  digitalWrite(9, HIGH);
  delay(FINAL); // Wait for 1000 millisecond(s)
  digitalWrite(9, LOW);
  delay(FINAL);
}

```

```
drive(L298N::MOTOR_A,HIGH,LOW);  
drive(L298N::MOTOR_B,HIGH,LOW);  
}  
void drive(int motor,int state1,int state2) {  
    driver.setup_motor(motor,state1,state2);  
    driver.drive_motor(motor,speed);  
}
```