

Program robot penginformasi kadar gas menggunakan sensor TGS dengan lintasan *wall follower*:

```
//==ultrasonic
#include <Ultrasonic.h>
Ultrasonic ultraDepan(5,53);
//Ultrasonic ultraKiri(6,-);
Ultrasonic ultraKanan(7,52);
int jarakDepan;
int jarakKanan;

//==SERIAL BLUETOOTH
#include <SoftwareSerial.h>
SoftwareSerial bt(17, 16); //Rx, Tx

//==LCD
#include <LiquidCrystal.h>
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

//==TGS
const int analogIn1 = A0;
const int analogIn2 = A2;
const int analogIn3 = A4;
int tgs1 = 0;
int tgs2 = 0;
```

```
int tgs3 = 0;
```

```
//==Motor
```

```
#define dir_kiri 47
```

```
#define pwm_kiri 2
```

```
#define pwm_kanan 3
```

```
#define dir_kanan 51
```

```
//==LED dan Buzzer
```

```
#define buzzer 31
```

```
#define merah 33
```

```
#define kuning 35
```

```
#define hijau 37
```

```
//==fuzzyLIB
```

```
#include <FuzzyRule.h>
```

```
#include <FuzzyComposition.h>
```

```
#include <Fuzzy.h>
```

```
#include <FuzzyRuleConsequent.h>
```

```
#include <FuzzyOutput.h>
```

```
#include <FuzzyInput.h>
```

```
#include <FuzzyIO.h>
```

```
#include <FuzzySet.h>
```

```
#include <FuzzyRuleAntecedent.h>
```

```

//==fuzzy
Fuzzy* fuzzy = new Fuzzy();
//==himpunanFuzzyINPUT
FuzzySet* dDekat = new FuzzySet(0, 0, 10, 10);
FuzzySet* dSedang = new FuzzySet(10, 10, 16, 16);
FuzzySet* dJauh = new FuzzySet(16, 16, 20, 20);

FuzzySet* kaDekat = new FuzzySet(0, 0, 10, 10);
FuzzySet* kaSedang = new FuzzySet(10, 10, 21, 21);
FuzzySet* kaJauh = new FuzzySet(21, 21, 30, 30);
//==himpunanFuzzyOUTPUT
FuzzySet* kiMundurPelan = new FuzzySet(-8, -8, 0, 0);
FuzzySet* kiMundurSedang = new FuzzySet(-13, -13, -8, -8);
FuzzySet* kiMajuPelan = new FuzzySet(0, 0, 8, 8);
FuzzySet* kiMajuSedang = new FuzzySet(8, 8, 13, 13);

FuzzySet* kaMundurPelan = new FuzzySet(-8, -8, 0, 0);
FuzzySet* kaMundurSedang = new FuzzySet(-13, -13, -8, -8);
FuzzySet* kaMajuPelan = new FuzzySet(0, 0, 8, 8);
FuzzySet* kaMajuSedang = new FuzzySet(8, 8, 13, 13);

void setup() {
// put your setup code here, to run once:
bt.begin(9600);
lcd.begin(16, 2);
lcd.print("ii");

```

```
pinMode(dir_kiri, OUTPUT);
pinMode(dir_kanan, OUTPUT);
motor(0,0);

//==LED dan Buzzer
pinMode(buzzer, OUTPUT); //buzzer
pinMode(merah, OUTPUT); //LED Merah
pinMode(kuning, OUTPUT); //LED Kuning
pinMode(hijau, OUTPUT); //LED Hijau
digitalWrite(buzzer, LOW);
digitalWrite(merah, LOW);
digitalWrite(kuning, LOW);
digitalWrite(hijau, LOW);

//==Deklarasi Variabel Fuzzy
//==INPUT
FuzzyInput* sensorDepan = new FuzzyInput(1);
sensorDepan->addFuzzySet(dDekat);
sensorDepan->addFuzzySet(dSedang);
sensorDepan->addFuzzySet(dJauh);
fuzzy->addFuzzyInput(sensorDepan);

FuzzyInput* sensorKanan = new FuzzyInput(2);
sensorKanan->addFuzzySet(kaDekat);
sensorKanan->addFuzzySet(kaSedang);
sensorKanan->addFuzzySet(kaJauh);
```

```
fuzzy->addFuzzyInput(sensorKanan);  
  
//==OUTPUT  
  
FuzzyOutput* spKiri = new FuzzyOutput(1);  
spKiri->addFuzzySet(kiMundurPelan);  
spKiri->addFuzzySet(kiMundurSedang);  
spKiri->addFuzzySet(kiMajuPelan);  
spKiri->addFuzzySet(kiMajuSedang);  
fuzzy->addFuzzyOutput(spKiri);
```

```
FuzzyOutput* spKanan = new FuzzyOutput(2);  
spKanan->addFuzzySet(kaMundurPelan);  
spKanan->addFuzzySet(kaMundurSedang);  
spKanan->addFuzzySet(kaMajuPelan);  
spKanan->addFuzzySet(kaMajuSedang);  
fuzzy->addFuzzyOutput(spKanan);
```

```
//==RULES
```

```
//1. KananDekat, DepanJauh = SerongKiriSedang
```

```
FuzzyRuleAntecedent* kaDekatDANdJauh = new FuzzyRuleAntecedent();  
kaDekatDANdJauh->joinWithAND(kaDekat, dJauh);
```

```
FuzzyRuleConsequent* SerongKiriSedang = new FuzzyRuleConsequent();  
SerongKiriSedang->addOutput(kiMajuPelan);  
SerongKiriSedang->addOutput(kaMajuSedang);
```

```
FuzzyRule*fuzzyRule1 = new FuzzyRule (1, kaDekatDANdJauh,  
SerongKiriSedang);
```

```
fuzzy->addFuzzyRule(fuzzyRule1);
```

```
//2. KananDekat, DepanSedang = SerongKiriPelan
```

```
FuzzyRuleAntecedent* kaDekatDANdSedang = new  
FuzzyRuleAntecedent();
```

```
kaDekatDANdSedang->joinWithAND(kaDekat, dSedang);
```

```
FuzzyRuleConsequent* SerongKiriPelan = new FuzzyRuleConsequent();
```

```
SerongKiriPelan->addOutput(kiMajuPelan);
```

```
SerongKiriPelan->addOutput(kaMajuSedang);
```

```
FuzzyRule* fuzzyRule2 = new FuzzyRule(2, kaDekatDANdSedang,  
SerongKiriPelan);
```

```
fuzzy->addFuzzyRule(fuzzyRule2);
```

```
//3. KananDekat, DepanDekat = MundurSedang
```

```
FuzzyRuleAntecedent* kaDekatDANdDekat = new FuzzyRuleAntecedent();
```

```
kaDekatDANdDekat->joinWithAND(kaDekat, dDekat);
```

```
FuzzyRuleConsequent* mundurSedang = new FuzzyRuleConsequent();
```

```
mundurSedang->addOutput(kiMundurSedang);
```

```
mundurSedang->addOutput(kaMundurSedang);
```

```
FuzzyRule* fuzzyRule3 = new FuzzyRule(3, kaDekatDANdDekat,  
mundurSedang);
```

```
fuzzy->addFuzzyRule(fuzzyRule3);
```

```
//4. KananSedang, DepanJauh = MajuSedang
```

```
FuzzyRuleAntecedent* kaSedangDANdJauh = new FuzzyRuleAntecedent();
```

```
kaSedangDANdJauh->joinWithAND(kaSedang, dJauh);
```

```
FuzzyRuleConsequent* majuSedang = new FuzzyRuleConsequent();
```

```
majuSedang->addOutput(kiMajuSedang);
```

```
majuSedang->addOutput(kaMajuSedang);
```

```
FuzzyRule* fuzzyRule4 = new FuzzyRule(4, kaSedangDANdJauh,  
majuSedang);
```

```
fuzzy->addFuzzyRule(fuzzyRule4);
```

```
//5. KananSedang, DepanSedang = BelokKiri
```

```
FuzzyRuleAntecedent* kaSedangDANdSedang = new  
FuzzyRuleAntecedent();
```

```
kaSedangDANdSedang->joinWithAND(kaSedang, dSedang);
```

```
FuzzyRuleConsequent* belokKiri = new FuzzyRuleConsequent();
```

```
belokKiri->addOutput(kiMundurSedang);
```

```
belokKiri->addOutput(kaMajuSedang);
```

```
FuzzyRule* fuzzyRule5 = new FuzzyRule(5, kaSedangDANdSedang,  
belokKiri);
```

```
fuzzy->addFuzzyRule(fuzzyRule5);
```

```
//6. KananSedang, DepanDekat = mundurPelan
```

```
FuzzyRuleAntecedent* kaSedangDANdDekat = new  
FuzzyRuleAntecedent();
```

```
kaSedangDANdDekat->joinWithAND(kaSedang, dDekat);
```

```
FuzzyRuleConsequent* mundurPelan = new FuzzyRuleConsequent();
```

```
mundurPelan->addOutput(kiMundurPelan);
```

```
mundurPelan->addOutput(kaMundurPelan);
```

```
FuzzyRule* fuzzyRule6 = new FuzzyRule(6, kaSedangDANdDekat,  
mundurPelan);
```

```
fuzzy->addFuzzyRule(fuzzyRule6);
```

```
//7. KananJauh, DepanJauh = SerongKananSedang
```

```
FuzzyRuleAntecedent* kaJauhDANdJauh = new FuzzyRuleAntecedent();
```

```
kaJauhDANdJauh->joinWithAND(kaJauh, dJauh);
```

```
FuzzyRuleConsequent* SerongKananSedang = new  
FuzzyRuleConsequent();
```

```
SerongKananSedang->addOutput(kiMajuSedang);
```

```
SerongKananSedang->addOutput(kaMajuPelan);
```

```
FuzzyRule* fuzzyRule7 = new FuzzyRule(7, kaJauhDANdJauh,  
SerongKananSedang);
```

```
fuzzy->addFuzzyRule(fuzzyRule7);
```

```
//8. KananJauh, DepanSedang = SerongKananPelan
```



```
FuzzyRuleAntecedent* kaJauhDANdSedang = new FuzzyRuleAntecedent();  
kaJauhDANdSedang->joinWithAND(kaJauh, dSedang);
```

```
FuzzyRuleConsequent* SerongKananPelan = new FuzzyRuleConsequent();  
SerongKananPelan->addOutput(kiMajuSedang);  
SerongKananPelan->addOutput(kaMajuPelan);
```

```
FuzzyRule* fuzzyRule8 = new FuzzyRule(8, kaJauhDANdSedang,  
SerongKananPelan);  
fuzzy->addFuzzyRule(fuzzyRule8);
```

```
//9. KananJauh, DepanDekat = MundurSedang, SerongKananPelan,  
FuzzyRuleAntecedent* kaJauhDANdDekat = new FuzzyRuleAntecedent();  
kaJauhDANdDekat->joinWithAND(kaJauh, dDekat);
```

```
FuzzyRuleConsequent* mundurPelan2 = new FuzzyRuleConsequent();  
mundurPelan2->addOutput(kiMundurPelan);  
mundurPelan2->addOutput(kaMundurPelan);
```

```
FuzzyRule* fuzzyRule9 = new FuzzyRule(9, kaJauhDANdDekat,  
mundurPelan2);
```

```
fuzzy->addFuzzyRule(fuzzyRule7);
```

```
}
```

```
void loop() {
```

```
while(1){  
  // put your main code here, to run repeatedly:  
  //loop_Ultrasonic  
  jarakDepan = ultraDepan.Ranging(CM);  
  jarakKanan = ultraKanan.Ranging(CM);  
  jarakDepan = constrain(jarakDepan, 0, 20);  
  jarakKanan = constrain(jarakKanan, 0, 30);  
  
  //loop_TGS  
  tgs1 = analogRead(analogIn1);  
  tgs2 = analogRead(analogIn2);  
  tgs3 = analogRead(analogIn3);  
  
  //print_tgs  
  bt.print("TGS 1 = ");  
  bt.print(tgs1);  
  bt.print(" ");  
  bt.print("TGS 2 = ");  
  bt.print(tgs2);  
  bt.print(" ");  
  bt.print("TGS 3 = ");  
  bt.print(tgs3);  
  bt.println(" ");
```

```
//LCD_tgs  
lcd.setCursor(0,0);  
lcd.print("|");  
lcd.print(tgs1);  
lcd.print("|");  
lcd.print("|");  
lcd.print(tgs2);  
lcd.print("|");  
lcd.print("|");  
lcd.print(tgs3);  
lcd.print("|");
```

```
//Print_Ultrasonic  
bt.print("depan:");  
bt.print(jarakDepan);  
bt.print(" cm");  
bt.print("kanan");  
bt.print(jarakKanan);  
bt.println(" cm");
```

```
//LCD_Ultrasonic  
lcd.setCursor(0,1);  
lcd.print("Ki:");  
lcd.print("0");  
lcd.print("De:");  
lcd.print(jarakDepan);
```

```
lcd.print("Ka:");
```

```
lcd.print(jarakKanan);
```

```
fuzzy->setInput(1, jarakDepan);
```

```
fuzzy->setInput(2, jarakKanan);
```

```
fuzzy->fuzzify();
```

```
float speedKi = fuzzy->defuzzify(1);
```

```
float speedKa = fuzzy->defuzzify(2);
```

```
speedKi = map(speedKi, -13, 13, -390, 130);
```

```
speedKa = map(speedKa, -13, 13, -390, 130);
```

```
motor(speedKi, speedKa);
```

```
bt.print("PWM_ki:");
```

```
bt.print(speedKi);
```

```
bt.print(" PWM_ka:");
```

```
bt.println(speedKa);
```

```
delay(100);
```

```
//==tgs1
```

```
if ((tgs1 >= 0) && (tgs1 <= 140)){
```

```
digitalWrite(buzzer, LOW);
```

```
digitalWrite(merah, LOW);
```

```
digitalWrite(kuning, LOW);
digitalWrite(hijau, HIGH);
}
else if ((tgs1 >= 141 && tgs1 <= 250)){
digitalWrite(buzzer, LOW);
digitalWrite(merah, LOW);
digitalWrite(kuning, HIGH);
digitalWrite(hijau, LOW);
}
else if ((tgs1 >= 251 && tgs1 <= 1023)){
digitalWrite(buzzer, HIGH);
digitalWrite(merah, HIGH);
digitalWrite(kuning, LOW);
digitalWrite(hijau, LOW);
break;
}

//==TGS2
if ((tgs2 >= 0) && (tgs2 <= 140)){
digitalWrite(buzzer, LOW);
digitalWrite(merah, LOW);
digitalWrite(kuning, LOW);
digitalWrite(hijau, HIGH);
}
else if ((tgs2 >= 141 && tgs2 <= 250)){
digitalWrite(buzzer, LOW);
```

```
digitalWrite(merah, LOW);
digitalWrite(kuning, HIGH);
digitalWrite(hijau, LOW);
}

else if ((tgs2 >= 251 && tgs2 <= 1023)){
digitalWrite(buzzer, HIGH);
digitalWrite(merah, HIGH);
digitalWrite(kuning, LOW);
digitalWrite(hijau, LOW);
break;
}

//==tgs3
if ((tgs3 >= 0) && (tgs3 <= 140)){
digitalWrite(buzzer, LOW);
digitalWrite(merah, LOW);
digitalWrite(kuning, LOW);
digitalWrite(hijau, HIGH);
}

else if ((tgs3 >= 141 && tgs3 <= 250)){
digitalWrite(buzzer, LOW);
digitalWrite(merah, LOW);
digitalWrite(kuning, HIGH);
digitalWrite(hijau, LOW);
}

else if ((tgs3 >= 251 && tgs3 <= 1023)){
```

```
digitalWrite(buzzer, HIGH);
digitalWrite(merah, HIGH);
digitalWrite(kuning, LOW);
digitalWrite(hijau, LOW);
break;
}

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

}
while(1){
  motor(0, 0);}
}
void motor(int sp_kiri, int sp_kanan) {
  if (sp_kiri >= 0) {
    digitalWrite(dir_kiri, LOW);
  }
  else {
    sp_kiri = 255 + sp_kiri;
    digitalWrite(dir_kiri, HIGH);
  }

  if (sp_kanan >= 0) {
    digitalWrite(dir_kanan, LOW);
  }
  else {
    sp_kanan = 255 + sp_kanan;
```

```
digitalWrite(dir_kanan, HIGH);  
}  
  
analogWrite(pwm_kiri, sp_kiri);  
analogWrite(pwm_kanan, sp_kanan);  
}
```