

LAMPIRAN

1. Hasil Pemrograman Fuzzy Logic pada Mikrokontroler

```
FINAL_PROGRAM §
// DEFINISI PIN ULTRASONIC //
#include <NewPing.h>
#define TRIG_PIN1 2
#define ECHO_PIN1 3
#define TRIG_PIN2 12
#define ECHO_PIN2 13
NewPing sensorKiri(TRIG_PIN1, ECHO_PIN1);
NewPing sensorDepan(TRIG_PIN2, ECHO_PIN2);
// END

// DEFINISI DRIVER MOTOR
#include <L298N.h>
const int ENA = 18; // motor A (kiri)
const int IN1 = 6;
const int IN2 = 7;
const int IN3 = 5;
const int IN4 = 8;
const int ENB = 9; // motor B (kanan)
L298N driver(ENA,IN1,IN2,IN3,IN4,ENB);
int time_delay = 100;
//END

#include <Fuzzy.h>
Fuzzy *fuzzy = new Fuzzy();

// SENSOR KIRI
FuzzySet *dekat = new FuzzySet(0,0,15,20);
FuzzySet *sedang = new FuzzySet(15,20,20,25);
FuzzySet *jauh = new FuzzySet(20,25,35,35);

// SENSOR DEPAN
FuzzySet *dekat2 = new FuzzySet(0,0,15,20);
FuzzySet *sedang2 = new FuzzySet(15,20,20,25);
FuzzySet *jauh2 = new FuzzySet(20,25,35,35);

// MOTOR KIRI
FuzzySet *berhenti1 = new FuzzySet(50,50,50,50);
FuzzySet *pelan = new FuzzySet(50,50,50,50);
FuzzySet *cepat = new FuzzySet(120,120,120,120);
// MOTOR KANAN
FuzzySet *berhenti2 = new FuzzySet(50,50,50,50);
FuzzySet *pelan2 = new FuzzySet(50,50,50,50);
FuzzySet *cepat2 = new FuzzySet(120,120,120,120);
void setup() {
  Serial.begin(9600);

  // Fuzzy Input sensor kiri
  FuzzyInput *kiri = new FuzzyInput(1);
  kiri->addFuzzySet(dekat);
  kiri->addFuzzySet(sedang);
  kiri->addFuzzySet(jauh);
  fuzzy->addFuzzyInput(kiri);

  // Fuzzy Input sensor depan
  FuzzyInput *depan = new FuzzyInput(2);
  depan->addFuzzySet(dekat2);
  depan->addFuzzySet(sedang2);
  depan->addFuzzySet(jauh2);
  fuzzy->addFuzzyInput(depan);

  // Fuzzy Output motor kiri
  FuzzyOutput *mKiri = new FuzzyOutput(1);
  mKiri->addFuzzySet(berhenti1);
  mKiri->addFuzzySet(pelan);
  mKiri->addFuzzySet(cepat);
  fuzzy->addFuzzyOutput(mKiri);

  // Fuzzy Output motor kanan
  FuzzyOutput *mKanan = new FuzzyOutput(2);
  mKanan->addFuzzySet(berhenti2);
  mKanan->addFuzzySet(pelan2);
  mKanan->addFuzzySet(cepat2);
  fuzzy->addFuzzyOutput(mKanan);

  // =====\
  // Penempatan input dan output sebagai berikut
  // "sensorKiri_sensorDepan"->joinWithAND(SENSOR KIRI,SENSOR DEPAN);

  // Rules
  FuzzyRuleAntecedent *dekat_dekat = new FuzzyRuleAntecedent();
  dekat_dekat->joinWithAND(dekat,dekat2);
  FuzzyRuleConsequent *c_b_1 = new FuzzyRuleConsequent();
  c_b_1->addOutput(cepat); // Motor kiri
  c_b_1->addOutput(berhenti2); // Motor kanan
  FuzzyRule *fuzzyRule1 = new FuzzyRule(1,dekat_dekat,c_b_1);
  fuzzy->addFuzzyRule(fuzzyRule1);
}
```

```

// Rules2
FuzzyRuleAntecedent *sedang_dekat = new FuzzyRuleAntecedent();
sedang_dekat->joinWithAND(sedang,dekat2);
FuzzyRuleConsequent *c_b_2 = new FuzzyRuleConsequent();
c_b_2 -> addOutput(cepat); // Motor kiri
c_b_2 -> addOutput(berhenti2); // Motor kanan
FuzzyRule *fuzzyRule2 = new FuzzyRule(2,sedang_dekat,c_b_2);
fuzzy->addFuzzyRule(fuzzyRule2);

// Rules3
FuzzyRuleAntecedent *jauh_dekat = new FuzzyRuleAntecedent();
jauh_dekat->joinWithAND(jauh,dekat2);
FuzzyRuleConsequent *c_b_3 = new FuzzyRuleConsequent();
c_b_3 -> addOutput(cepat); // Motor kiri
c_b_3 -> addOutput(berhenti2); // Motor kanan
FuzzyRule *fuzzyRule3 = new FuzzyRule(3,jauh_dekat,c_b_3);
fuzzy->addFuzzyRule(fuzzyRule3);

// Rule 4
FuzzyRuleAntecedent *dekat_sedang = new FuzzyRuleAntecedent();
dekat_sedang->joinWithAND(dekat,sedang2);
FuzzyRuleConsequent *c_p_1 = new FuzzyRuleConsequent();
c_p_1 -> addOutput(cepat); // Motor kiri
c_p_1 -> addOutput(pelan2); // Motor kanan v
FuzzyRule *fuzzyRule4 = new FuzzyRule(4,dekat_sedang,c_p_1);
fuzzy->addFuzzyRule(fuzzyRule4);

// Rule 5
FuzzyRuleAntecedent *sedang_sedang = new FuzzyRuleAntecedent();
sedang_sedang->joinWithAND(sedang,sedang2);
FuzzyRuleConsequent *c_c_1 = new FuzzyRuleConsequent();
c_c_1 -> addOutput(cepat); // Motor kiri
c_c_1 -> addOutput(cepat2); // Motor kanan v
FuzzyRule *fuzzyRule5 = new FuzzyRule(5,sedang_sedang,c_c_1);
fuzzy->addFuzzyRule(fuzzyRule5);

// Rule 6
FuzzyRuleAntecedent *jauh_sedang = new FuzzyRuleAntecedent();
jauh_sedang->joinWithAND(jauh,sedang2);
FuzzyRuleConsequent *p_c_1 = new FuzzyRuleConsequent();
p_c_1 -> addOutput(pelan); // Motor kiri
p_c_1 -> addOutput(cepat2); // Motor kanan
FuzzyRule *fuzzyRule6 = new FuzzyRule(6,jauh_sedang,p_c_1);
fuzzy->addFuzzyRule(fuzzyRule6);

// Rule 7
FuzzyRuleAntecedent *dekat_jauh = new FuzzyRuleAntecedent();
dekat_jauh->joinWithAND(dekat,jauh2);
FuzzyRuleConsequent *c_p_2 = new FuzzyRuleConsequent();
c_p_2 -> addOutput(cepat); // Motor kiri
c_p_2 -> addOutput(pelan2); // Motor kanan v
FuzzyRule *fuzzyRule7 = new FuzzyRule(7,dekat_jauh,c_p_2);
fuzzy->addFuzzyRule(fuzzyRule7);

// Rule 8
FuzzyRuleAntecedent *sedang_jauh = new FuzzyRuleAntecedent();
sedang_jauh->joinWithAND(sedang,jauh2);
FuzzyRuleConsequent *c_c_2 = new FuzzyRuleConsequent();
c_c_2 -> addOutput(cepat); // Motor kiri
c_c_2 -> addOutput(cepat3); // Motor kanan v
FuzzyRule *fuzzyRule8 = new FuzzyRule(8,sedang_jauh,c_c_2);
fuzzy->addFuzzyRule(fuzzyRule8);

// Rule 9
FuzzyRuleAntecedent *jauh_jauh = new FuzzyRuleAntecedent();
jauh_jauh->joinWithAND(jauh,jauh2);
FuzzyRuleConsequent *p_c_2 = new FuzzyRuleConsequent();
p_c_2 -> addOutput(pelan); // Motor kiri
p_c_2 -> addOutput(cepat3); // Motor kanan
FuzzyRule *fuzzyRule9 = new FuzzyRule(9,jauh_jauh,p_c_2);
fuzzy->addFuzzyRule(fuzzyRule9);
}

void loop() {
// int sensorDepan = 23;
// int sensorKiri = 18;
float sensorDepan = sensor2Depan.ping_cm();
float sensorKiri = sensorKiri.ping_cm();
float sensorDepanBckp;
float sensorKiriBckp;
if (sensorDepan != 0) sensorDepanBckp = sensorDepan;
if (sensorKiri != 0) sensorKiriBckp = sensorKiri;
if (sensorDepan == 0) sensorDepan = sensorDepanBckp;
if (sensorKiri == 0) sensorKiri = sensorKiriBckp;

fuzzy->setInput(1, sensorKiri);
fuzzy->setInput(2, sensorDepan);
fuzzy->fuzzify();

bool Rule1 = fuzzy->isFiredRule(1);
bool Rule2 = fuzzy->isFiredRule(2);
bool Rule3 = fuzzy->isFiredRule(3);
bool Rule4 = fuzzy->isFiredRule(4);
bool Rule5 = fuzzy->isFiredRule(5);
bool Rule6 = fuzzy->isFiredRule(6);
bool Rule7 = fuzzy->isFiredRule(7);
bool Rule8 = fuzzy->isFiredRule(8);
bool Rule9 = fuzzy->isFiredRule(9);

float motorKiri = fuzzy->defuzzify(1);
float motorKanan = fuzzy->defuzzify(2);

Serial.print("Sensor Kiri : ");
Serial.print(sensorKiri);
Serial.print(" ");
Serial.print("Sensor Depan : ");
Serial.println(sensorDepan);

Serial.print("Hasil Kiri Dekat : ");
Serial.print(dekat -> getPertinence());
Serial.print("\t Hasil Depan Dekat : ");
Serial.println(dekat2 -> getPertinence());
Serial.print("Hasil Kiri Sedang : ");
Serial.print(sedang -> getPertinence());
Serial.print("\t Hasil Depan Sedang : ");

```

```

Serial.println(sedang2 -> getPertinence());
Serial.print("Masuk Kiri Jauh : ");
Serial.print(jauh -> getPertinence());
Serial.print("Masuk Depan Jauh : ");
Serial.println(jauh2 -> getPertinence());

// CER RULE
Serial.print("Rule1 : ");
Serial.print(Rule1);
Serial.print("\nRule4 : ");
Serial.print(Rule4);
Serial.print("\nRule7 : ");
Serial.println(Rule7);
Serial.print("Rule2 : ");
Serial.print(Rule2);
Serial.print("\nRule5 : ");
Serial.print(Rule5);
Serial.print("\nRule8 : ");
Serial.println(Rule8);
Serial.print("Rule3 : ");
Serial.print(Rule3);
Serial.print("\nRule6 : ");
Serial.print(Rule6);
Serial.print("\nRule9 : ");
Serial.println(Rule9);
// END CER RULE
Serial.print("Motor Kiri : ");
Serial.println(MotorKiri);
Serial.print(" ");
Serial.print("Motor Kanan : ");
Serial.println(MotorKanan);

Serial.println("");

drive(L298N: MOTOR_A_HIGH, LOW, motorKiri); //RAJU
drive(L298N: MOTOR_B_LOW, HIGH, motorKanan);
delay(1000);

}

void drive(int motor, int state1, int state2, int spd) {
  driver.setup_motor(motor, state1, state2);
  if (motor == 1) {
    spd += 20;
  }
  driver.drive_motor(motor, spd);
}

```

2. Tampak wall follower robot





