APPENDIX

```
1.  /////////////////////////// SERVO ///////////////////////////
2.  #include <Servo.h>
3.  Servo myservo;        //creates servo object
4.  const int servo = 9;     //sets servo @pin 9
5.  int threshold = 100; //Sets the sensor threshold at 10 inches
6.  int angle =90; //Sets the Initial angle
7.
8.  /////////////////////////// INPUT ///////////////////////////
9.  int analogPin = A0;
10. float celcius;
11. int suhu;
12. float dingin, hangat, panas;
13.
14. const int echopin1 = 2;
15. const int trigpin1 = 3;
16. const int echopin2 = 4;
17. const int trigpin2 = 5;
18. long Rightduration, Leftduration;
19. float Rightdistance, Leftdistance;
20. float dekat, sedang, jauh;
21.
22. /////////////////////////// OUTPUT ///////////////////////////
23. int led = 6;
24. float kipas = 11;
25. float Padam, Redup, Terang, SangatTerang;
26. float Off, Lambat, Cepat, SangatCepat;
27. float rule00, rule01, rule02, rule10, rule11, rule12, rule20, rule21, rule22;
28. float pembilang, pembilang1;
29. float penyebut, penyebut1;
```

```
float out, out1;

////////////////////////// FUZZY INPUT //////////////////////////
void FuzzySuhu(){
  // Dingin
  if (celcius <= 25){
    dingin = 1;
  } else if (celcius >= 25 && celcius <= 30){
    dingin = (30-celcius)/(30-25);
  } else if (celcius >= 30){
    dingin = 0;
  }

  // Hangat
  if (celcius <= 25){
    hangat = 0;
  } else if (celcius >= 25 && celcius <= 30){
    hangat = (celcius-25)/(30-25);
  } else if (celcius >= 30 && celcius <= 35){
    hangat = (35-celcius)/(35-30);
  } else if (celcius >= 35){
    hangat = 0;
  }

  // Panas
  if (celcius <= 30 ){
    panas = 0;
  } else if (celcius >= 30 && celcius <= 35){
    panas = (celcius-30)/(35-30);
  } else if (celcius >= 35){
```

```
60.     panas = 1;
61.   }
62. }
63.
64. void FuzzyJarak(){
65.   // Dekat
66.   if (Rightdistance <= 20){
67.     dekat = 1;
68.   } else if (Rightdistance >= 20 && Rightdistance <= 60){
69.     dekat = (60-Rightdistance)/(60-20);
70.   } else if (Rightdistance >= 60){
71.     dekat = 0;
72.   }
73.
74.   // Sedang
75.   if (Rightdistance <= 20){
76.     sedang = 0;
77.   } else if (Rightdistance >= 20 && Rightdistance <= 60){
78.     sedang = (Rightdistance-20)/(60-20);
79.   } else if (Rightdistance >= 60 && Rightdistance <= 100){
80.     sedang = (100-Rightdistance)/(100-60);
81.   } else if (Rightdistance >= 100){
82.     sedang = 0;
83.   }
84.
85.   // Jauh
86.   if (Rightdistance <= 60 ){
87.     jauh = 0;
88.   } else if (Rightdistance >= 60 && Rightdistance <= 100){
89.     jauh = (Rightdistance-60)/(100-60);
```

```
90.    } else if (Rightdistance >= 100){
91.      jauh = 1;
92.    }
93.  }
94.
95.  ////////////////////// FUZZY OUTPUT //////////////////////
96.  void FuzzyKecepatan(){
97.    Off = 15;
98.    Lambat = 70;
99.    Cepat = 150;
100.  SangatCepat = 255;
101.}
102.
103.void FuzzyLED(){
104.  Padam = 5;
105.  Redup = 64;
106.  Terang = 130;
107.  SangatTerang = 255;
108.}
109.
110.void fuzzifikasi(){
111.  FuzzySuhu();
112.  FuzzyJarak();
113.  FuzzyKecepatan();
114.  FuzzyLED();
115.}
116.
117.void fuzzy_rule(){
118.  fuzzifikasi();
119.
```

```
  rule00 = min(dingin,dekat);
  rule01 = min(dingin,sedang);
  rule02 = min(dingin,jauh);

  rule10 = min(hangat,dekat);
  rule11 = min(hangat,sedang);
  rule12 = min(hangat,jauh);

  rule20 = min(panas,dekat);
  rule21 = min(panas,sedang);
  rule22 = min(panas,jauh);
}

void defuzzifikasi() {
  fuzzy_rule();

  //defuzzifikasi Kecepatan
  pembilang = ((rule00*Off) + (rule01*Off) + (rule02*Off) + (rule10*Lambat) + (rule11*Lambat) + (rule12*Cepat) + (rule20*Cepat) + (rule21*SangatCepat) + (rule22*SangatCepat));
  penyebut = (rule00 + rule01 + rule02 + rule10 + rule11 + rule12 + rule20 + rule21 + rule22);
  out = pembilang/penyebut;

  //defuzzifikasi LED
  pembilang1 = ((rule00*Padam) + (rule01*Padam) + (rule02*Padam) + (rule10*Redup) + (rule11*Redup) + (rule12*Terang) + (rule20*Terang) + (rule21*SangatTerang) + (rule22*SangatTerang));
  penyebut1 = (rule00 + rule01 + rule02 + rule10 + rule11 + rule12 + rule20 + rule21 + rule22);
  out1 = pembilang1/penyebut1;

```

```
146.  if(out1 < 64) {
147.    analogWrite(led, 0);
148.  }
149.
150.  if(out1 >= 64 && out1 < 130) {
151.    analogWrite(led, 5);
152.  }
153.
154.  if(out1 >= 130 && out1 < 255) {
155.    analogWrite(led, 30);
156.  }
157.
158.  if(out1 >= 255) {
159.    analogWrite(led, 255);
160.  }
161.}
162.
163.////////////////////////////////////////////////////////////
164.void setup()
165.{
166.  myservo.attach(9); //attaches the servo on pin 9
167.  Serial.begin(9600);
168.  pinMode(echopin1, INPUT);
169.  pinMode(trigpin1, OUTPUT);
170.  pinMode(echopin2, INPUT);
171.  pinMode(trigpin2, OUTPUT);
172.  pinMode(led, OUTPUT);
173.  pinMode(kipas, OUTPUT);
174.
175.  //Baca pin input
```

```
176.  suhu = analogRead(analogPin);
177.  celcius = (suhu*5.0*100.0)/1024.0;
178.}
179.
180.////////////////////////////////////////////////////////////
181.void loop()
182.{
183.  //this will convert the elapsed time into the distance
184.  Rightdistance = microsecondsToInches(Rightduration);
185.  Leftdistance = microsecondsToInches(Leftduration);
186.
187.  digitalWrite(trigpin2, LOW);
188.  delayMicroseconds(3);
189.  digitalWrite(trigpin2, HIGH);
190.  delayMicroseconds(5);
191.  digitalWrite(trigpin2, LOW);
192.  Rightduration = pulseIn(echopin2, HIGH); //reads high pulse
193.
194.  digitalWrite(trigpin1, LOW);  //mengatur trigpin1 untuk memberikan pulsa rendah
195.  delayMicroseconds(3);      //durasinya 3 mikrodetik
196.  digitalWrite(trigpin1, HIGH);   //mengatur trigpin1 untuk memberikan pulsa tinggi
197.  delayMicroseconds(5);     //durasinya 5 mikrodetik
198.  digitalWrite(trigpin1, LOW);  //mengatur trigpin1 untuk memberikan pulsa rendah
199.  Leftduration = pulseIn(echopin1, HIGH); //reads high pulse
200.
201.  follow();   //follows the movement
202.  tampilan();
203.}
204.
205.long microsecondsToInches(long microseconds) {
```

```
206.  // The speed of sound is 340 m/s or 73.746 microseconds per inch.
207.  // The ping travels out and back, so to find the distance of the
208.  // object we take half of the distance traveled.
209.  return microseconds / 74 / 2;
210.}
211.
212.void follow() //conditions for the follow command
213.{
214.  if (Rightdistance <= threshold || Leftdistance <= threshold)
215.  {
216.    if (Rightdistance + 30 < Leftdistance)
217.    {
218.      angle = angle - 30;
219.    }
220.    if (Leftdistance + 30 < Rightdistance)
221.    {
222.      angle = angle + 30;
223.    }
224.  }
225.
226.  if (angle > 180)
227.  {
228.    angle = 180;
229.  }
230.  if (angle < 0)
231.  {
232.    angle = 0;
233.  }
234.  myservo.write(angle);
235.}
```

```
236.
237.void tampilan() {
238.  Serial.print("Suhu :  ");
239.  Serial.print(celcius);
240.  Serial.print(" Celcius");
241.  Serial.print("  |  ");
242.
243.  Serial.print("Jarak Sensor Kanan : ");
244.  Serial.print(Rightdistance);
245.  Serial.print(" cm");
246.  Serial.print("  |  ");
247.
248.  Serial.print("Jarak Sensor Kiri : ");
249.  Serial.print(Leftdistance);
250.  Serial.println(" cm");
251.
252.  fuzzifikasi(); // memanggil fungsi fuzzifikasi untuk menghitung keanggotaan masing2
      variable
253.  fuzzy_rule(); // memanggil fungsi fuzzy rule
254.  defuzzifikasi(); // memanggil fungsi defuzzifikasi
255.
256.  Serial.print("dingin : ");
257.  Serial.print(dingin);
258.  Serial.print("  |  ");
259.  Serial.print("hangat : ");
260.  Serial.print(hangat);
261.  Serial.print("  |  ");
262.  Serial.print("panas : ");
263.  Serial.println(panas);
264.
265.  Serial.print("dekat : ");
```

```
266.  Serial.print(dekat);
267.  Serial.print("  |  ");
268.  Serial.print("sedang : ");
269.  Serial.print(sedang);
270.  Serial.print("  |  ");
271.  Serial.print("jauh : ");
272.  Serial.println(jauh);
273.
274.  Serial.print("Rule00 : ");
275.  Serial.print(rule00);
276.  Serial.print("  |  ");
277.  Serial.print("Rule01 : ");
278.  Serial.print(rule01);
279.  Serial.print("  |  ");
280.  Serial.print("Rule02 : ");
281.  Serial.println(rule02);
282.  Serial.print("Rule10 : ");
283.  Serial.print(rule10);
284.  Serial.print("  |  ");
285.  Serial.print("Rule11 : ");
286.  Serial.print(rule11);
287.  Serial.print("  |  ");
288.  Serial.print("Rule12 : ");
289.  Serial.println(rule12);
290.  Serial.print("Rule20 : ");
291.  Serial.print(rule20);
292.  Serial.print("  |  ");
293.  Serial.print("Rule21 : ");
294.  Serial.print(rule21);
295.  Serial.print("  |  ");
```

```
296.  Serial.print("Rule22 : ");
297.  Serial.println(rule22);
298.
299.  Serial.print("off : ");
300.  Serial.print(Off);
301.  Serial.print("  |  ");
302.  Serial.print("lambat : ");
303.  Serial.print(Lambat);
304.  Serial.print("  |  ");
305.  Serial.print("cepat : ");
306.  Serial.print(Cepat);
307.  Serial.print("  |  ");
308.  Serial.print("sangat cepat : ");
309.  Serial.println(SangatCepat);
310.
311.  Serial.print("padam : ");
312.  Serial.print(Padam);
313.  Serial.print("  |  ");
314.  Serial.print("redup : ");
315.  Serial.print(Redup);
316.  Serial.print("  |  ");
317.  Serial.print("terang : ");
318.  Serial.print(Terang);
319.  Serial.print("  |  ");
320.  Serial.print("sangat terang : ");
321.  Serial.println(SangatTerang);
322.
323.  Serial.print("Hasil DeFuzzy Kecepatan: ");
324.  Serial.print(out);
325.  Serial.print("  |  ");
```

```
326.  Serial.print("Hasil DeFuzzy LED: ");
327.  Serial.println(out1);
328.  Serial.println("");
329.  delay(5000);
330.}
```

PLAGIARISM CHECK.ORG

0.37% PLAGIARISM APPROXIMATELY

Report #14237487

CHAPTER 1 INTRODUCTION 1.1 Background In this project I want to solve the problem for fan users. Many people complain about the problem of expensive electricity costs. There are many possibilities that cause electricity costs to be expensive, one of the reasons is not turning off the fan when leaving the room. Most people don't turn off the fan when they leave the room because they forgot or were in a hurry. In addition, there are also problems that occur in some people who almost every day have to go back and forth to the fan just to get the right wind, such as when suddenly the situation in the room becomes very hot, making us have to stand up just to increase the fan speed. To solve this problem, I created a system using an LM35 sensor to detect temperature and an Ultrasonic sensor to determine the distance between the fan and the object. This system also uses the Fuzzy Sugeno algorithm to calculate the final value of the data obtained by the LM35 sensor and Ultrasonic sensor, so that the

m