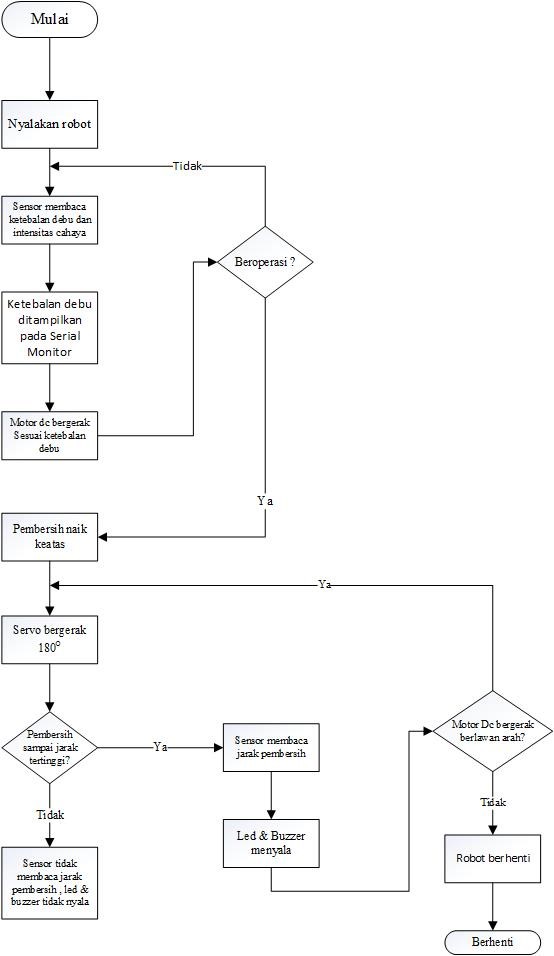
**PROTOTIPE ROBOT PEMBERSIH KACA**

1. **Alur Kerja Prototipe Robot Pembersih Kaca**

****

Robot pertama kali ditempelkan pada jendela, kemudian nyalakan robot. Setelah robot menyala  *dust* sensor (sensor debu) akan membaca ketebalan debu yang menempel pada permukaan kaca dan photodioda akan mendeteksi intensitas cahaya kemudian motor DC akan bergerak searah jarum jam dengan kecepatan sesuai ketebalan debu dan intensitas cahaya. Servo akan berputar 180o yang berfungsi sebagai penggerak spons untuk membersihkan kaca. Ketika pembersih sudah mencapai jarak tertinggi yang ditentukan maka sensor ultrasonik akan membaca jarak , kemudian led dan buzzer akan menyala sebagai indikator dan motor DC akan bergerak berlawanan arah jarum jam untuk menggerakkan kembali servo dan bagian pembersih ke posisi semula.

1. **Sketch Prototipe Robot Pembersih Kaca**

|  |  |
| --- | --- |
| FuzzyRobotPembersihKaca.ino | |
| #include <Arduino.h>    //sensor debu  const int pinGP2Y = A0;  int pinLED = 40;  int samplingTime = 280;  int deltaTime = 40;  int sleepTime = 9680;  int dustDensity;  int Set = 150;  //motor dc  const int IN1 = 7;  const int IN2 = 6;  const int ENA = 9;  //sensor ultrasonik  const int trigPin = 10;  const int echoPin = 11;  const int buzzer = 12;  const int ledPin = 13; | Pendefinisian Konstanta |
| //servo  #include <Servo.h>  Servo myservo; // membuat objek servo di library Servo.h  int pos = 0; // geser servopada posisi 0 derajat  //photodiode module  int photodiode = 42;  int ncahaya;  // fuzzy  #include <Fuzzy.h>  #include <FuzzyComposition.h>  #include <FuzzyIO.h>  #include <FuzzyInput.h>  #include <FuzzyOutput.h>  #include <FuzzyRule.h>  #include <FuzzyRuleAntecedent.h>  #include <FuzzyRuleConsequent.h>  #include <FuzzySet.h>  #define FUZZY\_IN\_DEBU 1  #define FUZZY\_IN\_CAHAYA 2  #define FUZZY\_OUT\_PWM 3  float DebuX;  float CahayaX; | Pendefinisian Variabel |
| #define APP\_PORT\_DEBUG Serial  // fuzzy object  Fuzzy \*fuzzy\_main\_obj = new Fuzzy();  // input debu  FuzzySet \*debu\_tipis = new FuzzySet(25, 25, 122.5, 220);  FuzzySet \*debu\_sedang = new FuzzySet(122.5, 220, 220, 330);  FuzzySet \*debu\_tebal = new FuzzySet(220, 330, 330, 440.18);  // input cahaya  FuzzySet \*cahaya\_gelap = new FuzzySet(37, 37, 143.5, 250);  FuzzySet \*cahaya\_agak\_terang = new FuzzySet(143.5, 250, 250, 281);  FuzzySet \*cahaya\_terang = new FuzzySet(250, 281, 281, 312);  // output pwm  FuzzySet \*pwm\_lambat = new FuzzySet(100, 100, 125, 150);  FuzzySet \*pwm\_sedang = new FuzzySet(125, 150, 150, 202.5);  FuzzySet \*pwm\_cepat = new FuzzySet(150, 202.5, 202.5, 255); | Pembuatan Fuzzy Object |
| void APP\_DEBUG\_PRINT(String alog) {  char dtx[16] = {0};  snprintf\_P(dtx, sizeof(dtx), (const char \*)F("%-10u : "), millis());  APP\_PORT\_DEBUG.println(String(dtx) + alog);  }  FuzzyRule \*createNewFuzzyRule(int ruleId, FuzzySet \*in1, FuzzySet \*in2, FuzzySet \*out1) {  FuzzyRuleConsequent \*fzThen = new FuzzyRuleConsequent();  fzThen->addOutput(out1);  FuzzyRuleAntecedent \*fzIf = new FuzzyRuleAntecedent();  fzIf->joinWithAND(in1, in2);  return new FuzzyRule(ruleId, fzIf, fzThen);  }  /\*\*  \* init all fuzzy input, rule, and output  \* @method fuzzyInit  \*/  void fuzzyInit() {  // FuzzyInput DEBU  FuzzyInput \*fz\_DEBU = new FuzzyInput(FUZZY\_IN\_DEBU);  fz\_DEBU->addFuzzySet(debu\_tipis);  fz\_DEBU->addFuzzySet(debu\_sedang);  fz\_DEBU->addFuzzySet(debu\_tebal);  fuzzy\_main\_obj->addFuzzyInput(fz\_DEBU);  // FuzzyInput CAHAYA  FuzzyInput \*fz\_CAHAYA = new FuzzyInput(FUZZY\_IN\_CAHAYA);  fz\_CAHAYA->addFuzzySet(cahaya\_gelap);  fz\_CAHAYA->addFuzzySet(cahaya\_agak\_terang);  fz\_CAHAYA->addFuzzySet(cahaya\_terang);  fuzzy\_main\_obj->addFuzzyInput(fz\_CAHAYA);  // FuzzyOutput PWM  FuzzyOutput \*fz\_PWM = new FuzzyOutput(FUZZY\_OUT\_PWM);  fz\_PWM->addFuzzySet(pwm\_lambat);  fz\_PWM->addFuzzySet(pwm\_sedang);  fz\_PWM->addFuzzySet(pwm\_cepat);  fuzzy\_main\_obj->addFuzzyOutput(fz\_PWM);  // fuzzy rule  fuzzy\_main\_obj->addFuzzyRule(  createNewFuzzyRule(1, debu\_tipis, cahaya\_gelap, pwm\_lambat));  fuzzy\_main\_obj->addFuzzyRule(  createNewFuzzyRule(2, debu\_tipis, cahaya\_agak\_terang, pwm\_lambat));  fuzzy\_main\_obj->addFuzzyRule(  createNewFuzzyRule(3, debu\_tipis, cahaya\_terang, pwm\_cepat));  fuzzy\_main\_obj->addFuzzyRule(  createNewFuzzyRule(4, debu\_sedang, cahaya\_gelap, pwm\_lambat));  fuzzy\_main\_obj->addFuzzyRule(  createNewFuzzyRule(5, debu\_sedang, cahaya\_agak\_terang, pwm\_sedang));  fuzzy\_main\_obj->addFuzzyRule(  createNewFuzzyRule(6, debu\_sedang, cahaya\_terang, pwm\_cepat));  fuzzy\_main\_obj->addFuzzyRule(  createNewFuzzyRule(7, debu\_tebal, cahaya\_gelap, pwm\_lambat));  fuzzy\_main\_obj->addFuzzyRule(  createNewFuzzyRule(8, debu\_tebal, cahaya\_agak\_terang, pwm\_lambat));  fuzzy\_main\_obj->addFuzzyRule(  createNewFuzzyRule(9, debu\_tebal, cahaya\_terang, pwm\_cepat));  }  void fuzzyProcessInput(float DebuX, float CahayaX, float \*pwmX) {  fuzzy\_main\_obj->setInput(FUZZY\_IN\_DEBU, DebuX);  fuzzy\_main\_obj->setInput(FUZZY\_IN\_CAHAYA, CahayaX);  fuzzy\_main\_obj->fuzzify();  \*pwmX = fuzzy\_main\_obj->defuzzify(FUZZY\_OUT\_PWM);  } | Pembuatan Fuzzy Rule |
| void debugTest() {  float PwmX = 255;  //float debu  float voMeasured = analogRead(pinGP2Y);  delayMicroseconds(deltaTime);  digitalWrite(pinLED, HIGH);  delayMicroseconds(sleepTime);  float calcVoltage = voMeasured \* (3.3 / 1024);  float dustDensity = (0.17 \* calcVoltage - 0.1) \* 1000;  if (dustDensity < 0) {  dustDensity = 0;  }  int ncahaya = analogRead(photodiode);    float DEBU = abs (dustDensity);  float CAHAYA = abs (ncahaya);    // set timeout for serial  APP\_PORT\_DEBUG.setTimeout(5);  APP\_DEBUG\_PRINT(("PEMBACAAN SENSOR CAHAYA = ") + String(CAHAYA));  APP\_DEBUG\_PRINT(("PEMBACAAN SENSOR DEBU = ") + String(DEBU));  float DebuX = DEBU;  float CahayaX = CAHAYA;    if (DebuX > 0){  fuzzyProcessInput(DebuX, CahayaX, &PwmX);  }else{  PwmX = 0;  }  // hasil defuzikasi  APP\_DEBUG\_PRINT(String("DEBU FUZZY = ") + String(DebuX) + String(" -- ") +  String(debu\_tipis->getPertinence()) + String(" -- ") +  String(debu\_sedang->getPertinence()) + String(" -- ") +  String(debu\_tebal->getPertinence()));  APP\_DEBUG\_PRINT(String("CAHAYA FUZZY = ") + String(CahayaX) + String(" -- ") +  String(cahaya\_gelap->getPertinence()) + String(" -- ") +  String(cahaya\_agak\_terang->getPertinence()) + String(" -- ") +  String(cahaya\_terang->getPertinence()));    Serial.print("PWM = ");  Serial.print(PwmX);  } | Pembuatan kelas debugTest() |
| void setup() {  APP\_PORT\_DEBUG.begin(9600);  //fuzzy  Serial.println("Log Fuzzy Logic Control"); //Print a message    //sensor debu  pinMode(pinLED,OUTPUT);  //motor dc  pinMode (IN1, OUTPUT);  pinMode (IN2, OUTPUT);  pinMode (ENA, OUTPUT);  //sensor ultrasonik  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output  pinMode(echoPin, INPUT); // Sets the echoPin as an Input  pinMode(buzzer, OUTPUT);  pinMode(ledPin, OUTPUT);  //servo  myservo.attach(31); //definisikan pin yang digunakan untuk mengontrol motor servo adalah pin 31  //photodiode module  pinMode(photodiode, OUTPUT);  // init fuzzy  fuzzyInit();  APP\_DEBUG\_PRINT(F("INIT DONE"));  } | Pembuatan kelas void setup() |
| void loop() {  //sensor debu  digitalWrite(pinLED,LOW); // power on the LED  delayMicroseconds(samplingTime);  float voMeasured = analogRead(pinGP2Y);  delayMicroseconds(deltaTime);  digitalWrite(pinLED, HIGH);  delayMicroseconds(sleepTime);  float calcVoltage = voMeasured \* (3.3 / 1024);  float dustDensity = (0.17 \* calcVoltage - 0.1) \* 1000;  if (dustDensity < 0) {  dustDensity = 0;  }  //photodiode modul  int ncahaya = analogRead(photodiode);  //motor dc  analogWrite(ENA, 255);  digitalWrite(IN1, LOW);  delay(500);  digitalWrite(IN2, HIGH);  delay(500);  //sensor ultrasonik  digitalWrite(trigPin, LOW);  delayMicroseconds(1);  digitalWrite(trigPin, HIGH);  delayMicroseconds(1);  digitalWrite(trigPin, LOW);  duration = pulseIn(echoPin, HIGH);  distance= (duration/2) / 29.1; // Calculating the distance  float DebuX = dustDensity;  float CahayaX = ncahaya;    safetyDistance = distance;  if (safetyDistance < 10 && safetyDistance > 1){  digitalWrite(buzzer, HIGH);  digitalWrite(ledPin, HIGH);  analogWrite(ENA, 255);    digitalWrite(IN1, HIGH);  delay(500);  digitalWrite(IN2, LOW);  delay(500);  }  else if  (safetyDistance < 20 && safetyDistance > 10){    analogWrite(ENA, 255);    digitalWrite(IN1, HIGH);  delay(500);  digitalWrite(IN2, LOW);  delay(500);  }  else  {  digitalWrite(buzzer, LOW);  digitalWrite(ledPin, LOW);  analogWrite(ENA, 255);    digitalWrite(IN1, LOW);  delay(500);  digitalWrite(IN2, HIGH);  delay(500);  }  //servo  for (pos = 0; pos <= 180; pos += 1) { // menjalankan fungsi for loop dari 0 - 180 dan nilai ini akan digunakan sebagai nilai posisi servo  myservo.write(pos); // mengatur posisi servo berdasarkan nilai dari for loop  delay(5); // beri jeda 5 miliseconds untuk setiap perubahan posisi  }  for (pos = 180; pos >= 0; pos -= 1) { // menjalankan fungsi for loop dari 180 - 0 dan nilai ini akan digunakan sebagai nilai posisiservo  myservo.write(pos); // mengatur posisi servo berdasarkan nilai dari for loop  delay(5); // beri jeda 5 miliseconds untuk setiap perubahan posisi  }    float PwmX = 255;    float DEBU = dustDensity;  float CAHAYA = ncahaya;    debugTest();  DebuX = DEBU;  CahayaX = CAHAYA;  fuzzyProcessInput(DebuX, CahayaX, &PwmX);    } | Pembuatan Kelas void loop() |