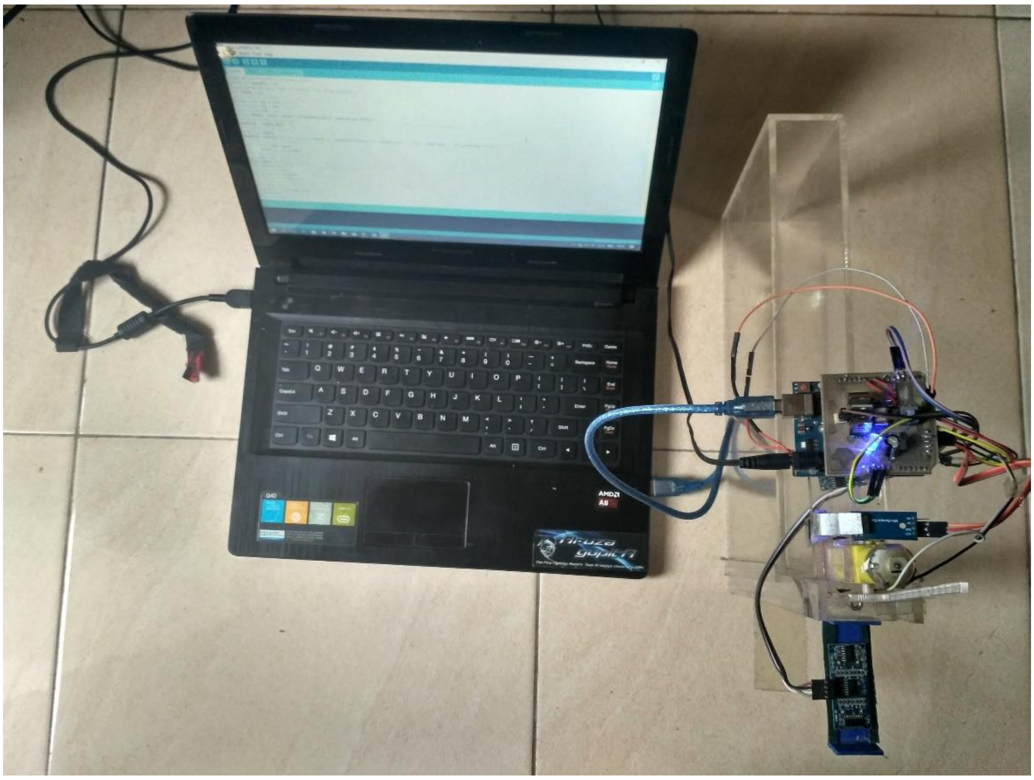
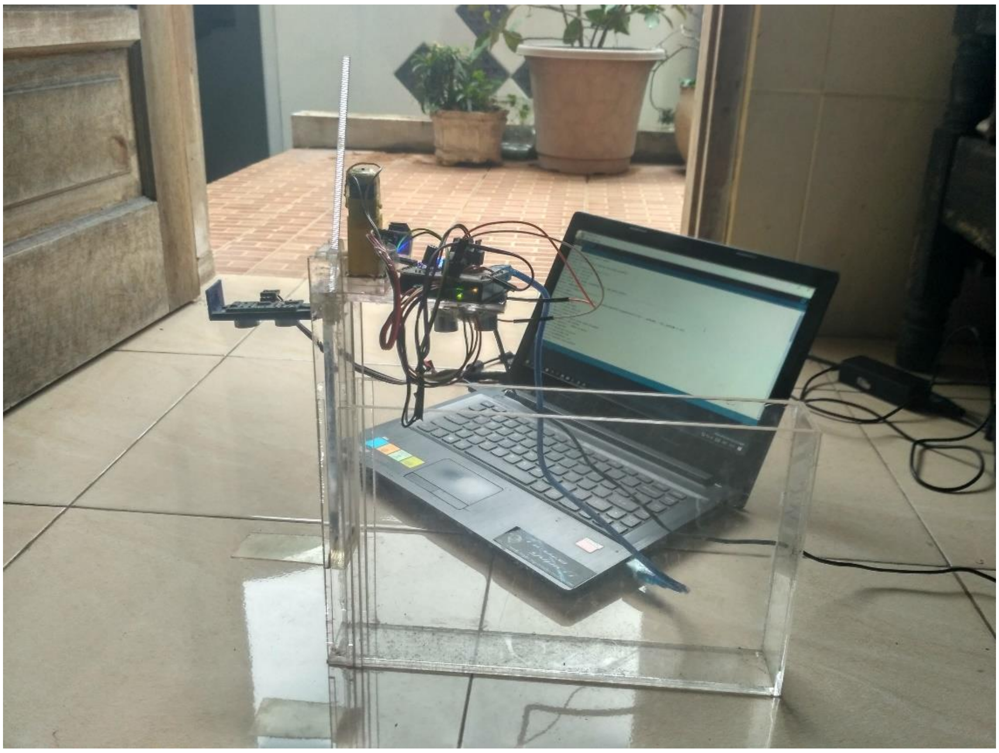
LAMPIRAN I

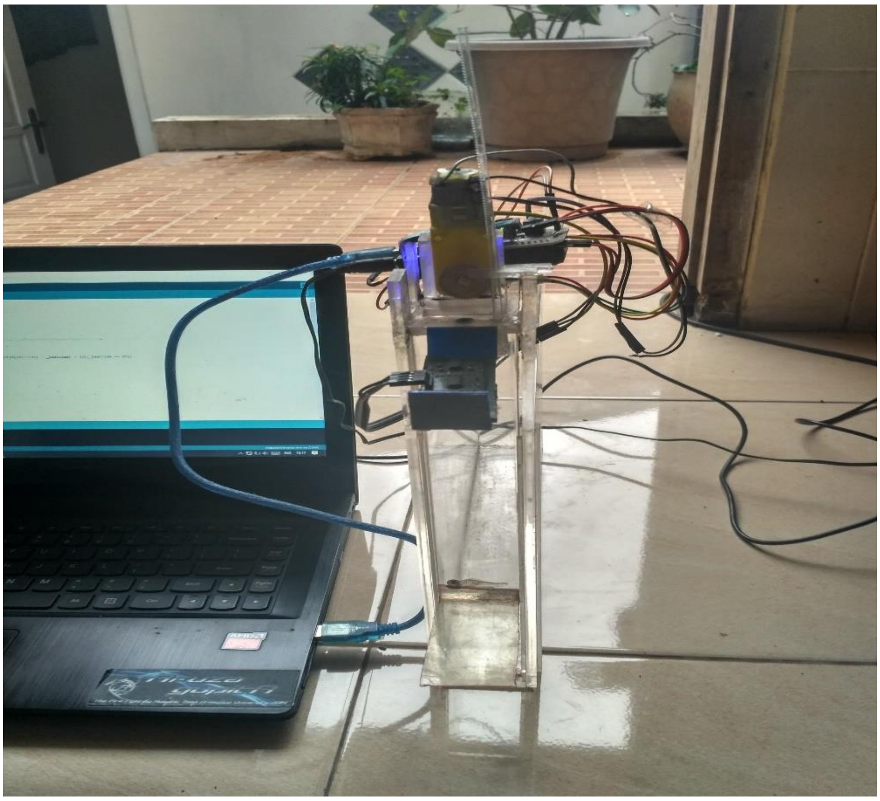
Foto Alat



Gambar Plant tampak atas



Gambar Plant tampak samping



Gambar Plant tampak depan

LAMPIRAN II

Listing Program

Arduino Program

```
/**********************/
/* YOGA ADHIYASA. */
/**********************/

//pin which triggers ultrasonic sound
const int pingPin = 4;
//pin which delivers time to receive echo using pulseIn()
int inPin = 3;
/////////////////PID///////////////////////
const float Kp = 14,4;
const float Ki = 1,8;
const float Kd = 28,8;
float pTerm, iTerm, dTerm, integrated_error, last_error, error;
const float K = 1.4;
#define   GUARD_GAIN   5
/////////////////Timer////////////////////
uint32_t timer;
#define runEvery(t) for (static typeof(t) _lasttime;(typeof(t))((typeof(t))millis() - _lasttime) > (t);_lasttime += (t))
/////////////////variabel/////////////////
int speed, hasil,jadi;
float duration, su,tinggi;
int dir = 10;//driver motor dc
int rem = 9;//driver motor dc
int limmit=7;//limmit switch
int encoder_pin = 2; // pulse output from the module
unsigned int rpm=0; // rpm reading
volatile byte pulses=0; // number of pulses
unsigned long timeold=0;
// number of pulses per revolution
// based on your encoder disc
unsigned int pulsesperturn = 12;
int jarak=6;// tentukan jarak setpoint
int setRPM=72;
void counter()
{
  //Update count
  pulses++;
}
// TODO: Make calibration routine
void analogWrite25k(int pin, int value)
{
   switch (pin) {
       case 9:
           OCR1A = value;
           break;
       case 10:
           OCR1B = value;
           break;
       default:
           // no other pin will work
           break;
   }
```

```
}
void setup() {
   Serial.begin(9600);
  // Configure Timer 1 for PWM @ 25 kHz.
   //TCCR1B = TCCR1B & B11111000 | B00000001; // Set PWM frequency for D9 & D10
:
    TCCR1A = 0;           // undo the configuration done by...
    TCCR1B = 0;           // ...the Arduino core library
    TCNT1  = 0;           // reset timer
    TCCR1A = _BV(COM1A1)  // non-inverted PWM on ch. A
           | _BV(COM1B1)  // same on ch; B
           | _BV(WGM11);  // mode 10: ph. correct PWM, TOP = ICR1
    TCCR1B = _BV(WGM13)   // ditto
           | _BV(CS10);   // prescaler = 1
    ICR1   = 320;         // TOP = 320
  pinMode(pingPin, OUTPUT);
  pinMode(inPin, INPUT);
  pinMode(limmit, INPUT);
  pinMode(dir,OUTPUT);
  pinMode(rem,OUTPUT);
  pinMode(encoder_pin, INPUT);
   //Interrupt 0 is digital pin 2
   //Triggers on Falling Edge (change from HIGH to LOW)
  attachInterrupt(0, counter, FALLING);
   // Initialize
  // pulses = 0;
   rpm = 0;
   timeold = 0;
}

void loop() {
  //runEvery(25){
//int a;
//a=digitalRead(limmit);
   sensorU();
      hasil=270;
      Rotary();
      Pid();
      Motors();
    if (rpm<setPWM){speed=speed+1;}
  Serial.print("RPM = ");
  Serial.println(rpm,DEC);

   Serial.print("tinggi = ");
   Serial.print(tinggi);
   Serial.println("");

   Serial.print("Speed = ");
   Serial.println(speed);

}

//////Motor//////
void Motors(){
  if (speed > 0)
  {
    //forward
    speed = map(speed,0,-255,0,255);
    analogWrite25k(rem, speed);
    analogWrite25k(dir, 0);
  }
  else
  {
    // backward
    speed = map(speed,0,-255,0,255);
    analogWrite25k(rem, 0);
    analogWrite25k(dir, speed);
  }
}

void stop(){
  analogWrite25k(rem, 0);
  analogWrite25k(dir, 0);
}
///////PID///////////
void Pid(){
  error= tinggi-jarak;
  pTerm = Kp * error;
  integrated_error += error;
  iTerm = Ki * constrain(integrated_error, -GUARD_GAIN, GUARD_GAIN);
  dTerm = Kd * (error - last_error);
  last_error = error;
  speed = constrain(K*(pTerm + iTerm + dTerm), -255, 255);
  }
void Rotary(){
  if (millis() - timeold >= 1000) {
      //Don't process interrupts during calculations
      detachInterrupt(0);
      rpm = (72 * 1000 / pulsesperturn )/ (millis() - timeold)* pulses;
      timeold = millis();
      pulses = 0;
      jadi=rpm/3;
      attachInterrupt(0, counter, FALLING);
    }
  }
///////Ultrasonic////////
void sensorU(){
  //initializing the pin states
  //sending the signal, starting with LOW for a clean signal
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  duration = pulseIn(inPin, HIGH);

  // convert the time into a distance
  su = microsecondsToCentimeters(duration);
  tinggi=22.88-su;
  //printing the current readings to ther serial display
  //Serial.print(su);
  //Serial.print("cm");
  //Serial.println();

  delay(100);
  }
float microsecondsToCentimeters(float microseconds)
{
  return microseconds / 29 / 2;
}
```

LAMPIRAN III

Datasheet