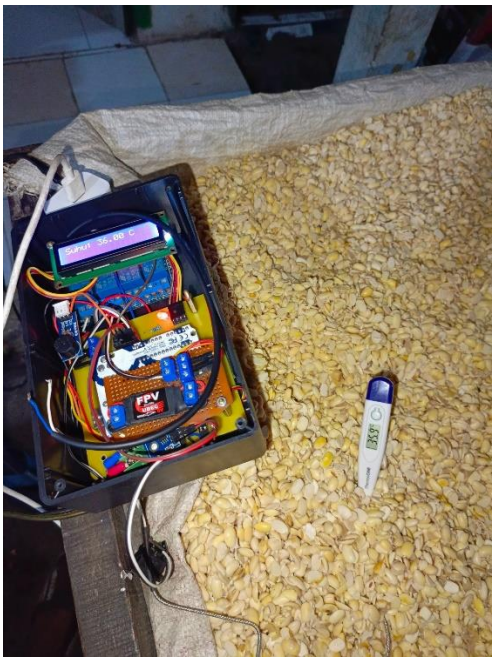
`

LAMPIRAN

Lampiran 1. Hasil uji coba alat

1. pengujian thermocouple terhadap thermometer one alpha one



Gambar 1 pengujian thermocouple

Gambar diatas adalah proses pengujian thermocouple terhadap thermometer

2. pengujian waktu terhadap kecepatan





A B

Gambar 2. Penguji terhadap waktu kecepatan (A),(B)

`

Lampiran 3 proses hasil keseluruhan alat

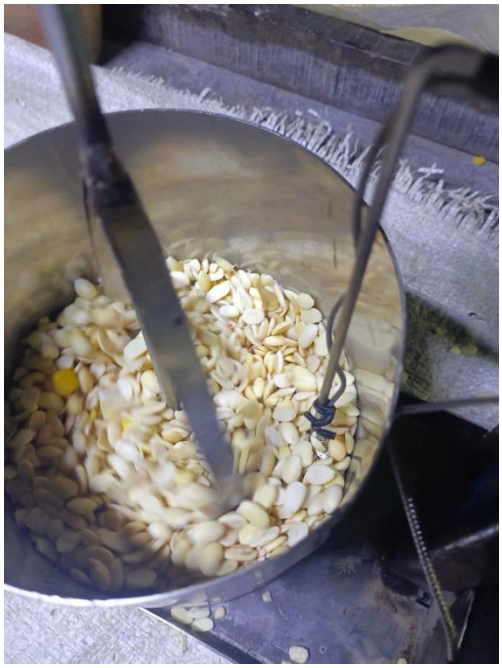








`





Lampiran 4 pengujian pendinginan kedelai mengunakan kipas Dc



`

Lampiran 5 pengujian Rpm menggunakan Tachometer



Lampiran 5 program dari alat

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <max6675.h>

#define Motor_IN1 2
#define Motor_IN2 5
#define Motor_ENA 6
#define Buzzer 8
#define Button 3 // Define the push button pin
#define Relay 4  // Define the relay pin
```

`

```
// Pin untuk MAX6675
int thermoDO = 12;
int thermoCS = 10;
int thermoCLK = 13;

// Inisialisasi MAX6675
MAX6675 thermocouple(thermoCLK, thermoCS, thermoDO);

// Inisialisasi LCD dengan alamat I2C 0x27
LiquidCrystal_I2C lcd(0x27, 16, 2); // Sesuaikan alamat I2C jika diperlukan

float tempOffset = 0.0; // Variabel untuk menyimpan offset kalibrasi suhu
int motorSpeed = 255;   // Variabel untuk menyimpan kecepatan motor (0-255)

unsigned long motorOnStartTime = 0; // Waktu saat motor dinyalakan
bool motorOn = false; // Status motor

void setup() {
  // Memulai komunikasi serial untuk debugging
  Serial.begin(9600);

  // Memulai LCD
  lcd.init();
  lcd.backlight();
  lcd.print("-Selamat Datang-");
  delay(2000);
  lcd.clear();
```

```
  lcd.print("--Proyek Akhir--");
  delay(2000);
  lcd.clear();
  lcd.print("Membaca sensor");
  delay(2000);
  lcd.clear();

  // Mengatur pin sebagai output atau input
  pinMode(Motor_IN1, OUTPUT);
  pinMode(Motor_IN2, OUTPUT);
  pinMode(Motor_ENA, OUTPUT);
  pinMode(Buzzer, OUTPUT);
  pinMode(Button, INPUT_PULLUP); // Setup button pin
  pinMode(Relay, OUTPUT); // Setup relay pin

  // Menampilkan instruksi kalibrasi pada Serial Monitor
  Serial.println("Gunakan perintah 'setOffset nilai' untuk mengatur offset kalibrasi suhu.");
  Serial.println("Contoh: setOffset 2.5");
  Serial.println("Gunakan perintah 'setSpeed nilai' untuk mengatur kecepatan motor (0-255).");
  Serial.println("Contoh: setSpeed 128");
}

void loop() {
  digitalWrite(Motor_IN1, LOW);
  digitalWrite(Motor_IN2, LOW);
  analogWrite(Motor_ENA, 0);
```

```
  digitalWrite(Buzzer, HIGH);
  digitalWrite(Relay, LOW); // Ensure the relay is off initially

  // Membaca suhu dari sensor thermocouple
  float t = thermocouple.readCelsius() + tempOffset; // Sesuaikan suhu dengan offset kalibrasi
  // Memeriksa apakah pembacaan gagal
  if (isnan(t)) {
    Serial.println("Gagal membaca dari sensor Thermocouple!");
    lcd.clear();
    lcd.print("Kesalahan Thermo");
    delay(2000); // Menunggu 2 detik sebelum mencoba lagi
    return;
  }

  // Mencetak suhu ke Serial Monitor
  Serial.print("Suhu: ");
  Serial.print(t);
  Serial.println(" *C");

  // Menampilkan suhu di LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Suhu: ");
  lcd.print(t);
  lcd.print(" C");
  delay(5000);
```

```
lcd.clear();

if (t <= 34.0) {
  digitalWrite(Motor_IN1, LOW); // Turn off the motor
  digitalWrite(Motor_IN2, LOW);
  analogWrite(Motor_ENA, 0);
  digitalWrite(Relay, HIGH); // Turn on the relay to start cooling

  // Buzzer beeping effect
  for (int i = 0; i < 10; i++) { // Beep 10 times
    digitalWrite(Buzzer, LOW); // Turn on the buzzer
    delay(200); // Beep duration
    digitalWrite(Buzzer, HIGH); // Turn off the buzzer
    delay(200); // Pause between beeps
  }

  lcd.setCursor(0, 0);
  lcd.print("Motor OFF");
  lcd.setCursor(0, 1);
  lcd.print("Tambahkan ragi");

  // Waiting for the push button to be pressed
  while (digitalRead(Button) == HIGH) {
    delay(100); // Debounce delay
  }

  digitalWrite(Motor_IN1, HIGH); // Turn on the motor
```

`

```
  digitalWrite(Motor_IN2, LOW);
  analogWrite(Motor_ENA, motorSpeed);
  motorOnStartTime = millis(); // Record the start time
  motorOn = true;

  lcd.clear();
  lcd.print("Mengaduk kedelai");
  lcd.setCursor(0, 1);
  lcd.print("Motor ON");
 }

if (motorOn) {
  if (millis() - motorOnStartTime >= 180000) { // Check if 180 seconds have passed
    digitalWrite(Motor_IN1, LOW); // Turn off the motor
    digitalWrite(Motor_IN2, LOW);
    analogWrite(Motor_ENA, 0);
    motorOn = false; // Reset the motor status

    // Scroll text "---SELESAI---" continuously
    String selesai = "----SELESAI-----";
    while (true) {
      for (int position = 0; position < selesai.length(); position++) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(selesai.substring(position) + selesai.substring(0, position));
        delay(300); // Delay to control the speed of scrolling
      }
```

`

```
      // Buzzer beeping effect in scrolling
      for (int i = 0; i < 5; i++) { // Beep 5 times
        digitalWrite(Buzzer, LOW); // Turn on the buzzer
        delay(200); // Beep duration
        digitalWrite(Buzzer, HIGH); // Turn off the buzzer
        delay(200); // Pause between beeps
      }
    }
  }
 } else {
  digitalWrite(Motor_IN1, LOW); // Ensure the motor is off
  digitalWrite(Motor_IN2, LOW);
  analogWrite(Motor_ENA, 0);
  digitalWrite(Buzzer, HIGH); // Ensure the buzzer is off
  digitalWrite(Relay, HIGH); // Ensure the relay is off
 }

// Mengecek apakah ada input dari Serial Monitor untuk kalibrasi
if (Serial.available() > 0) {
  String input = Serial.readStringUntil('\n');
  input.trim(); // Menghapus spasi di awal dan akhir string

  if (input.startsWith("setOffset ")) {
    input = input.substring(10); // Menghapus "setOffset " dari input
    tempOffset = input.toFloat(); // Mengonversi sisa string ke float
    Serial.print("Offset kalibrasi suhu diatur ke: ");
```

```
      Serial.println(tempOffset);
    } else if (input.startsWith("setSpeed ")) {
      input = input.substring(9); // Menghapus "setSpeed " dari input
      motorSpeed = input.toInt(); // Mengonversi sisa string ke int
      Serial.print("Kecepatan motor diatur ke: ");
      Serial.println(motorSpeed);
    }
  }
}
```