

LAPORAN PROYEK AKHIR

CLOSED-LOOP SYSTEM CONTROL FOR A WATER HEATER WITH MAGNETIC STIRRER USING ARDUINO

Dosen Pengampu : Mahmudah Salwa Gianti, S.Si., M.Eng., Diky Zakaria, S.Pd. M.T.,
Muhammad Rizalul Wahid, S.Si., M.T.



disusun oleh :

Ardita Aura Syawaliani	2307386
Eko Purwanto	2306254
Muhammad Iqbal Ridho	2309141
Muhammad Miftah Fadhilah Al-Ghifari	2308678
Rizkhy Anugrah Putra Setiawan	2312264
Sunan Maulana Sulinda Dwika Darma	2312131

**PROGRAM STUDI MEKATRONIKA DAN KECERDASAN
BUATAN KAMPUS DAERAH PURWAKARTA UNIVERSITAS
PENDIDIKAN INDONESIA
2024**

DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR GAMBAR	ii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan Proyek	2
1.4 Batasan Masalah.....	2
BAB 2 LANDASAN TEORI.....	3
2.1 Mikrokontroler	3
2.2 Sistem Kendali	3
2.3 Motor Listrik	3
BAB 3 METODOLOGI	5
3.1 Desain Sistem.....	5
3.2 Komponen dan Alat	6
3.3 Langkah Kerja.....	7
3.4 Pengujian Sistem.....	8
BAB 4 Hasil dan Pembahasan.....	9
4.1 Hasil Proyek	9
4.2 Kendala yang Dihadapi	11
4.3 Solusi Kendala	12
4.5 Analisis Sistem.....	13
BAB 5 KESIMPULAN DAN SARAN.....	14
5.1 Kesimpulan	14
5.2 Saran.....	15
DAFTAR PUSTAKA	16
LAMPIRAN	17

DAFTAR GAMBAR

Gambar 1. Flowchart sistem water heater.....	5
Gambar 2. Flowchart cara kerja sistem magnetic stirrer.....	5
Gambar 3. Diagram closed loop sistem water heater.....	6
Gambar 4. Tampilan GUI	9
Gambar 5. Tampilan Hasil Proyek tampak kiri	10
Gambar 6. Tampilan hasil proyek tampak kanan.....	10
Gambar 7. Wiring pemasangan sensor LDR.....	12
Gambar 8. Wiring diagram sistem water heater dan magnetic stirrer.....	28

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Proyek ini awalnya dibuat untuk mata kuliah Sistem Kendali untuk mempraktekan sistem kontrol closed loop yaitu prototipe sistem water heater yang cara kerjanya menaikkan suhu air berdasarkan set point yang diatur dan tetap menjaganya dengan bantuan sensor. Kemudian karena digabung dengan mata kuliah mikrokontroler dan motor listrik maka kami menambahkan sistem magnetic stirrer atau pengaduk otomatis yang pengaduknya menggunakan magnet, sehingga tujuan proyek ini menjadi pemanas air dan pengaduk otomatis untuk membuat minuman.

Adanya proyek ini diharapkan bisa memberikan kemudahan dan solusi inovatif bagi para pelaku Usaha Mikro, Kecil, dan Menengah (UMKM) yang bergerak di bidang minuman. Proyek ini diharapkan dapat mempermudah mereka untuk menjual minuman dan mengimplementasikan sistem otomatis atau sistem closed loop ke jualannya. Dengan mengaplikasikan proyek ini, penjual diharapkan dapat meningkatkan kualitas layanan dan mendukung usaha mereka agar dapat bersaing di era digital.

Proyek ini bisa dikembangkan tidak hanya untuk menjadi pemanas pembuat minuman, tetapi bisa untuk aplikasi yang lebih luas dan berskala besar di berbagai sektor seperti water heater untuk kebutuhan rumah tangga, pasteurisasi susu untuk memastikan keamanan dan kualitas susu sebelum didistribusikan ke pasar, dan untuk industri pabrik yang membutuhkan suhu air tertentu seperti pengolahan bahan kimia atau makanan. Selain itu, proyek ini juga dapat menjadi alat bantu di laboratorium kimia untuk membantu peneliti ketika eksperimen dan penelitian yang membutuhkan pengendali suhu dan magnetic stirrer.

1.2 Rumusan Masalah

- Apakah proyek ini berhasil mengimplementasikan sistem kendali closed loop?
- Apakah proyek ini berhasil mengimplementasikan motor listrik?
- Apakah proyek ini berhasil mengimplementasikan mikrokontroler?

1.3 Tujuan Proyek

- Mempelajari lebih dalam dan mempraktekkan sistem kendali closed loop.
- Mempelajari lebih dalam dan mempraktekan mikrokontroler sebagai otak dalam sistem kami.
- Mempelajari lebih dalam dan mempraktekkan motor listrik ke dalam proyek kami.

1.4 Batasan Masalah

- Proyek ini menggunakan sistem kendali closed loop dengan kendali on-off.
- Proyek ini menggunakan mikrokontroler Arduino Uno R3 ATmega328P.
- Proyek ini menggunakan motor dc 3-6V dan motor servo.

BAB 2 LANDASAN TEORI

2.1 Mikrokontroler

Mikrokontroler yang kami gunakan adalah Arduino Uno R3 yaitu mikrokontroler berbasis ATmega328P yang mempermudah pengembangan sistem elektronik. Mikrokontroler ini memiliki memori Flash sebesar 32 KB, SRAM 2 KB, EEPROM 1 KB, dan bekerja pada frekuensi clock 16 MHz dengan tegangan operasi 5V. Arduino Uno R3 memiliki 14 pin digital I/O (6 dapat digunakan sebagai PWM) dan 6 pin analog input, yang mendukung berbagai aplikasi seperti membaca sensor dan mengontrol aktuator. (Feri Djuandi, 2011).

Pemrograman Arduino Uno R3 dilakukan melalui Arduino IDE menggunakan bahasa C/C++, dengan struktur dasar berupa fungsi `setup()` untuk inisialisasi dan `loop()` untuk proses berulang. Mikrokontroler ini menggunakan arsitektur Harvard RISC yang efisien, dengan dukungan regulator tegangan untuk stabilitas daya. (Feri Djuandi, 2011).

2.2 Sistem Kendali

Sistem kendali yang kami gunakan adalah sistem kontrol closed loop on-off. Sistem kendali jenis ini menggunakan feedback untuk mengontrol keluaran agar sesuai dengan set point yang diatur. Dalam sistem ini sensor digunakan mendeteksi output dan mengirimkan data tersebut ke pengontrol dan kemudian membandingkan nilai keluaran asli dengan nilai referensi atau setpoint dan memberikan sinyal untuk menghidupkan dan mematikan aktuator tergantung pada kondisi yang terdeteksi. (Golnaraghi, Fardi. Kuo, 2010).

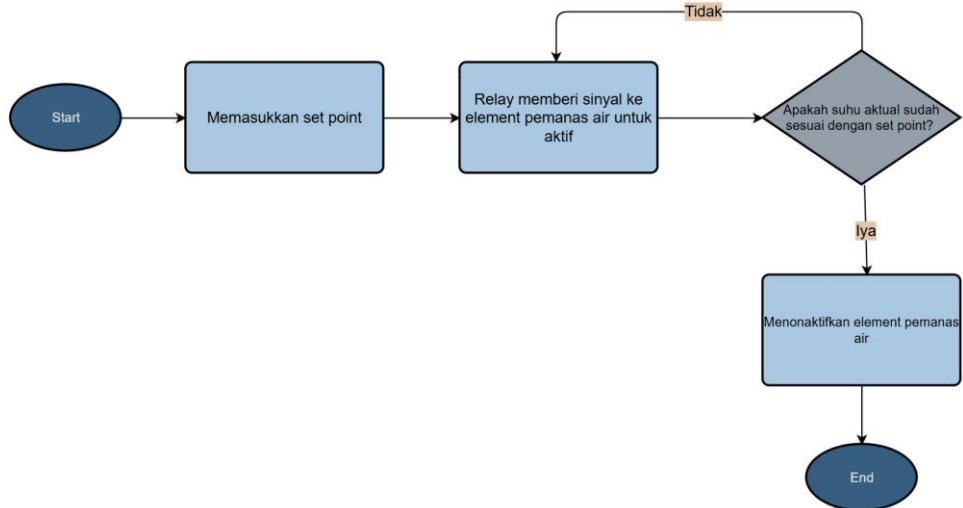
2.3 Motor Listrik

Motor listrik yang kami gunakan adalah motor DC 3-6V yang menggunakan PWM dan motor servo. Motor DC adalah aktuator yang mengubah energi listrik menjadi energi mekanik atau gerak. Motor ini bekerja dari rentang 3-6 volt dan memiliki dua terminal yaitu positif dan negatif. Motor DC ini bisa diatur kecepatannya menggunakan PWM (Pulse Width Modulation) yang mengatur daya rata-rata yang diberikan kepada motor. (Setiawan et al., 2017).

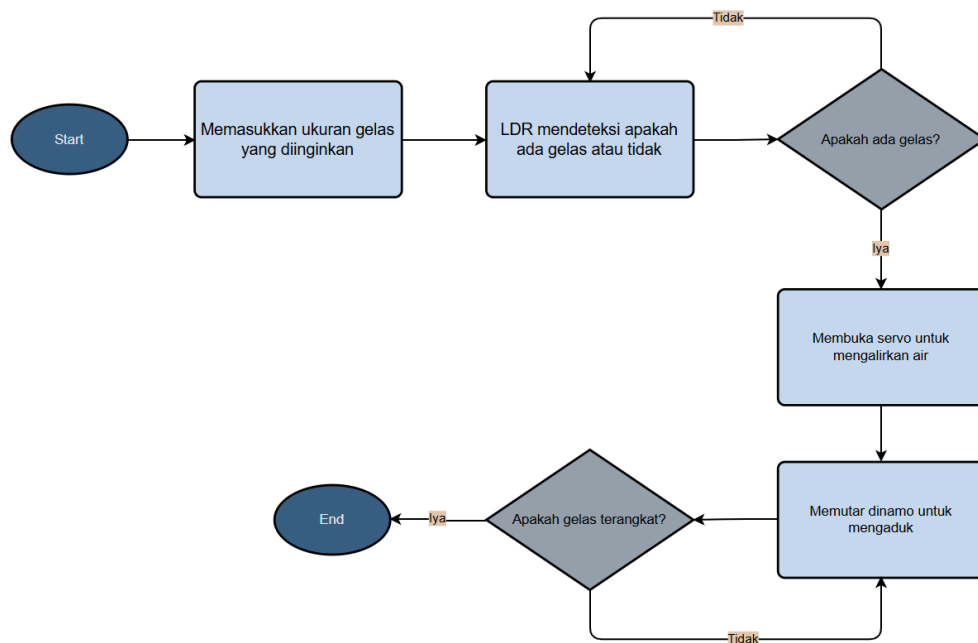
Motor servo adalah aktuator yang dibuat untuk kontrol yang presisi terhadap posisi. Motor ini biasanya terdiri dari motor brushless, gearbox, IC. Rentang sudut yang bisa dicapai oleh motor servo adalah 0° - 180° yang diatur dengan sinyal atau pulsa yang dikirim ke motor servo. (Hilal & Manan, 2015).

BAB 3 METODOLOGI

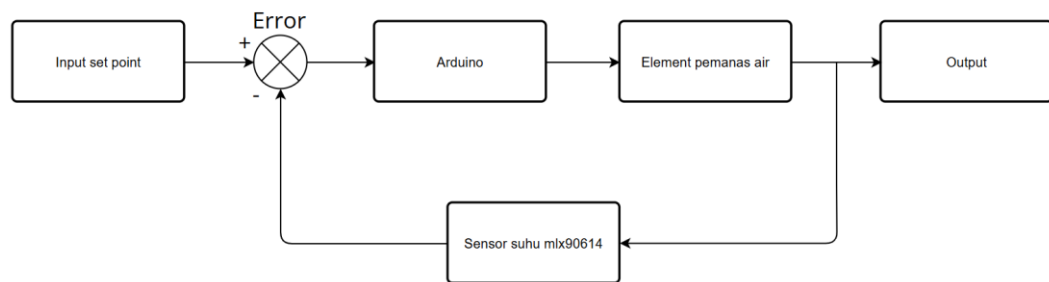
3.1 Desain Sistem



Gambar 1. Flowchart sistem water heater



Gambar 2. Flowchart cara kerja sistem magnetic stirrer



Gambar 3. Diagram closed loop sistem water heater

3.2 Komponen dan Alat

a. Software

- Arduino IDE
- Thonny
- Visual Studio Code

b. Komponen

- Arduino Uno R3
- Breadboard
- Kabel jumper
- Kabel 1.5mm
- Relay SSR-25DA
- Resistor 1k Ω
- Driver motor mx1508
- Sensor LDR
- Motor servo MG90
- Motor DC 3-6V
- Sensor suhu mlx90614
- Element pemanas
- Male-female banana jack
- Magnet stirrer
- Magnet neodmium

c. Alat

- Solder
- Cutter

- Tang pemotong
 - Tang kupas kabel
 - Lem tembak
 - Lem korea
 - Obeng
 - termometer
- d. Bahan
- Wadah 2.5L
 - Wadah toples 300ml
 - Impra board
 - Stik es krim

3.3 Langkah Kerja

- Water heater
 1. Buat box untuk menaruh wadah air 2.5L-nya dan magnetic stirrer dan menyimpan breadboard dan arduino di dalamnya.
 2. Buat bracket untuk sensor mlx90614 menghadap ke wadah untuk mendeteksi suhu air.
 3. Buat bracket untuk menaruh heater yang akan memanaskan air.
 4. Buat bracket untuk motor servo yang akan membuka katup/keran-nya nanti untuk mengalirkan air ke gelasnya.
 5. Sambungkan pin input servo ke pin digital 8 dan sambungkan vcc dan groundnya.
 6. Sambungkan pin sensor mlx90614 pin SCL ke A5 dan SDA ke A4 arduino kemudian pin vcc dan groundnya.
 7. Sambungkan kabel fasa heater ke relay SSR-25DA dan bagian input ke pin digital 7 dan ke ground.
- Magnetic Stirrer
 1. Siapkan wadah toples 300ml dan diwarnai hitam agar nantinya pembacaan sensor LDR tidak terganggu.
 2. Lubangi bagian bawah toples untuk nantinya jalur keluar kabel motor DC 3-6V dan sensor LDR.

3. Buat bracket untuk motor DC-nya bisa menghadap ke atas/berdiri.
4. Buat bracket untuk sensor LDR yang tempatnya di atas motor DC untuk mendeteksi apakah ada gelas atau tidak.
5. Buat bracket untuk menempatkan magnet neodimium yang akan jadi pengaduknya.
6. Buat rangkaian pembagi tegangan dengan resistor agar sensor LDR bisa berjalan kemudian sambungkan pin sensor LDR ke pin A0 pada arduino.
7. Sambungkan pin motor DC 3-6V ke pin digital 11 dan 10 melalui motor driver mx1508 terlebih dahulu yang motor driver-sudah disambungkan dahulu pin vcc dan groundnya.

3.4 Pengujian Sistem

Metode pengujian yang kami terapkan adalah uji eksperimental, yaitu kami membandingkan suhu aktual dengan suhu yang terbaca di sensor, kemudian mengkalibrasi ulang pada sistem berdasarkan error sehingga membuat sistem berjalan berdasarkan keadaan yang riil. Alat ukur yang kami gunakan sebagai penguji suhu yang aktual adalah digital food thermometer.

BAB 4 Hasil dan Pembahasan

4.1 Hasil Proyek

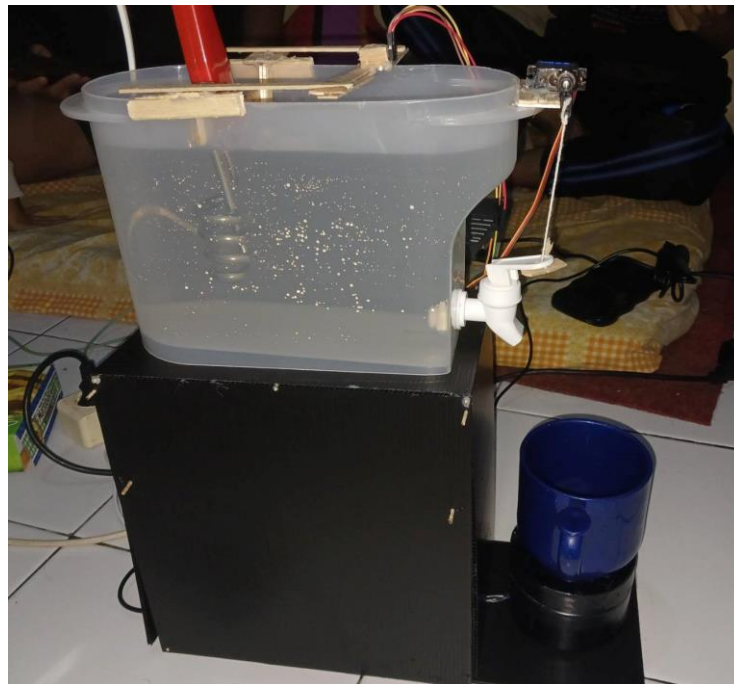
Sistem water heater dan magnetic stirrer ini bekerja dengan mengatur set-point suhu yang diinginkan, kemudian sensor MLX90614 akan mengecek apakah suhu air yang ada pada dispenser di bawah suhu set-point, jika iya maka element pemanas akan aktif dan ketika suhu sudah sesuai dengan set-point maka element pemanas akan mati, kemudian jika suhu air sudah di bawah set-point, element pemanas akan menyala kembali. Sedangkan magnetic stirrer bekerja jika sensor LDR mendeteksi adanya gelas di atas magnetic stirrer, kemudian servo akan menyala dan membuka katup dispenser sesuai dengan debit air yang diinginkan. Setelah air dituang ke gelas, motor DC akan berputar dan magnet yang ada pada gelas akan mengaduk dengan kecepatan yang diinginkan.

Berikut adalah link youtube demonstrasinya:

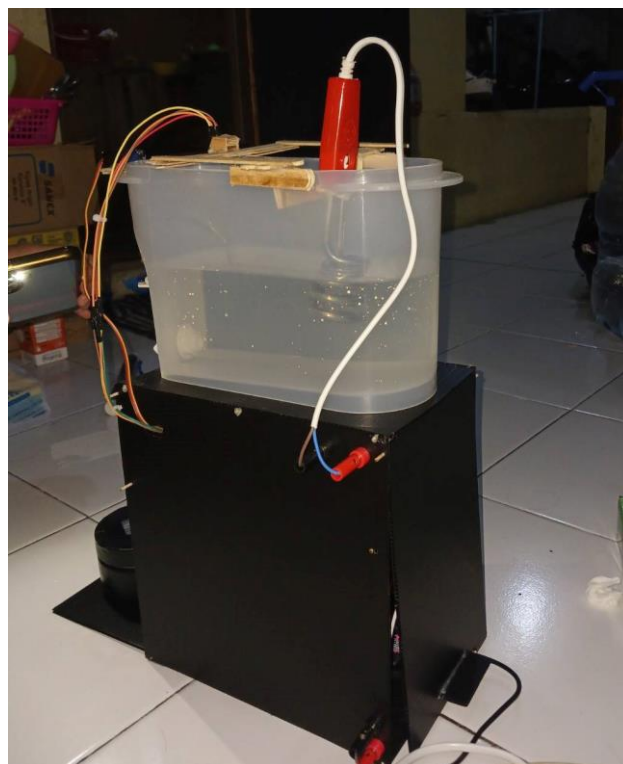
https://youtu.be/iEf8lo_XbI?si=p7zFJHPOXO0r7t0V



Gambar 4. Tampilan GUI



Gambar 5. Tampilan Hasil Proyek tampak kiri



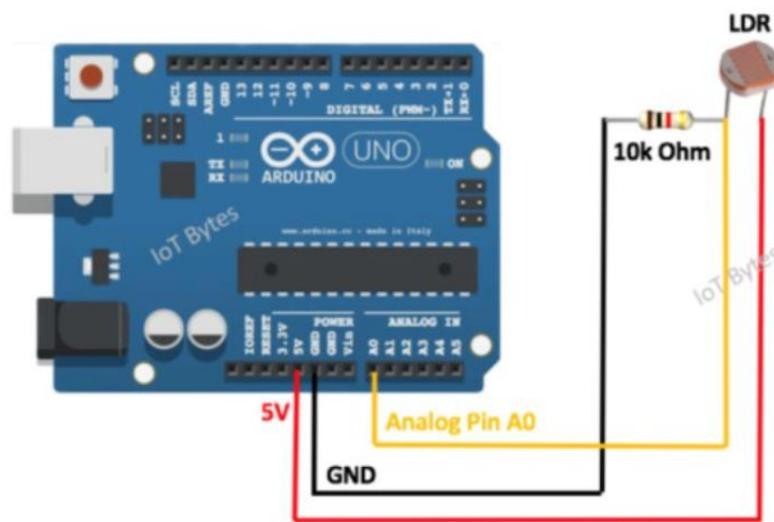
Gambar 6. Tampilan hasil proyek tampak kanan

4.2 Kendala yang Dihadapi

1. Nilai sensor LDR yang terbaca pada serial monitor keliru, ketika sensor ditutup atau mendeteksi bahwa sedang gelap, nilai sensor LDR bernilai tinggi sedangkan ketika sensor LDR diberi cahaya atau mendeteksi sedang terang maka nilai sensor LDR bernilai rendah.
2. Ketika mencoba motor DC 3-6V langsung ke arduino maka tidak bisa berjalan motor DC nya dan tidak bisa dimanipulasi kecepatan motornya.
3. Ketika kami mencoba menggabungkan dua aktuator motor servo dan motor DC untuk menjalankan suatu sistem magnetic stirrer, motor servo berhasil berjalan sedangkan motor DC tidak bisa berjalan.
4. Kami menggunakan motor servo untuk membuka katup/keran untuk mengalirkan air ke gelas, tetapi ketika kami mencobanya motor servo hanya kuat mengangkan sedikit katup/kerannya.
5. Sensor LDR sempat selalu membaca "1023" terus di serial monitor.
6. Kami menggunakan wadah untuk magnetic stirrer-nya adalah toples transparan dan terdapat sensor LDR di dalamnya untuk mendeteksi terdapat gelas atau tidak di atas nya, karena kami memakai toples bening maka masih banyak cahaya yang bisa masuk ke dalam toples sehingga sensor LDR kesulitan mendeteksi bahwa sudah ada gelas di atasnya atau tidak.

4.3 Solusi Kendala

1. Kami salah pemasangan VCC dan groundnya, karena cara kerja sensor LDR adalah dengan pembagi tegangan dengan resistor jadi VCC disambungkan dengan kaki sensor LDR dan ground disambungkan ke salah satu kaki resistor. Saat itu kami VCC dan groundnya kebalik menjadikan pembacaan sensor LDR ketika gelap nilai LDR menjadi tinggi nilainya sedangkan ketika terang nilai LDR menjadi rendah nilainya. Solusinya adalah memastikan kembali pemasangan VCC dan groundnya tepat terpasangnya.



Gambar 7. Wiring pemasangan sensor LDR

2. Solusinya adalah dengan menggunakan motor driver, kami menggunakan motor driver jenis mx1508 sehingga motor baru bisa berjalan. Dengan menggunakan motor driver juga kami jadi bisa memanipulasi kecepatan motor-nya dengan memberi PWM kepada motor tersebut.
3. Motor servo dan motor DC tidak bisa berjalan dikarenakan konflik timer, motor DC menggunakan input pin 9 dan 10 di arduino awalnya dan motor servo menggunakan input pin 8. Setelah dicari tahu ternyata pin 9 dan 10 adalah timer 1 pada arduino, dan pemasangan input servo yaitu pin 8 dapat mengganggu timer 1 pada arduino sehingga terjadi konflik timer. Solusinya adalah dengan mengubah input pin motor DC yang tadinya pin 9 dan 10 menjadi pin 10 dan 11 setelah itu baru motor servo dan motor DC baru bisa berjalan.

4. Awalnya kami menggunakan motor servo SG90 untuk membuka keran/katupnya tetapi ternyata tidak kuat untuk sampai membuka keran/katupnya sehingga kami menggantinya dengan motor servo jenis MG90. Perbedaan motor servo SG90 dan MG90 adalah pada kekuatan beban yang diangkatnya, SG90 hanya mampu sampai 1.4kg bebannya sedangkan MG90 mampu sampai 2.5kg. saat kami menggunakan motor servo MG90 juga kami menghadapi kendala yaitu housing/dudukan as gear-nya patah dan akhirnya kami memakai housing as gear dari motor servo yang lama.
5. Kendala tersebut terjadi karena kami terlalu dalam memasukkan kabel jumper yang terhubung dari sensor LDR menuju VCC pada breadbord, jadi kami memperbaiki sambungannya dengan memasukkan kabel jumper pada kedalaman yang sesuai.
6. Karena wadah toples yang kami pakai transparan dan mempengaruhi pembacaan sensor LDR sehingga sensor terus-menerus mendapat intensitas cahaya yang tinggi, kami melapisi toples dengan lakban hitam hingga menutup 90% permukaan toples dan hanya menyisakan sebagian kecil permukaan toples yang tidak dilapisi lakban hitam agar sensor LDR tetap dapat membaca intensitas cahaya di luar toples.

4.5 Analisis Sistem

1. Terdapat perbedaan suhu pada bagian atas permukaan air dan bagian bawah air setelah diukur menggunakan termometer disebabkan element pemanas air yang tidak sepenuhnya memasankan seluruh bagaian air di wadah. Perbedaaan suhu pada bagian atas dan bagian bawah ini menyebabkan pembcaan sensor tidak merata karena sensor hanya membaca suhu pada permukaan air.
2. Pembacaan nilai sensor LDR yang tergantung keadaan tempat, sehingga jika digunakan di tempat/area yang redup sensor akan mendeteksi bahwa terdapat gelas dan menyebabkan motor servo menyala dan membuka katup/keran dan mengalirkan air.

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan praktikum yang telah dilakukan dapat ditarik kesimpulan bahwa sistem kontrol closed berhasil diimplementasikan dengan element pemanas berhasil menjaga suhu air dengan sistem on-off yang dibantu relay SSR-25DA dan dengan bantuan feedback dari sensor mlx90614. Sistem magnetic stirrer juga berhasil dijalankan dengan bantuan sensor LDR untuk mendeteksi adanya gelas di atas magnetic stirrer. Sistem magnetic stirrer ini dapat diatur kecepatannya dengan mengatur PWM sesuai yang diinginkan. Sistem pemanas air dan magnetic stirrer dapat dijalankan dengan adanya mikrokontroler Arduino Uno R3 ATmega328P. Semua sistem ini dapat dikontrol dengan Graphical User Interface (GUI).

Setelah melakukan praktikum ini kami mendapatkan banyak ilmu tentang mikrokontroler khususnya Arduino Uno R3 seperti dapat terjadinya konflik timer dan menghubungkan Arduino ke GUI dengan Python seperti menampilkan grafik suhu ke GUI.

Kami berhasil mengimplementasikan materi mata kuliah Sistem Kendali, Motor Listrik, dan Mikrokontroler yang kami pelajari 1 semester ini. Kami memilih menggunakan sistem kendali closed loop jenis kendali on-off dan GUI untuk mengimplementasikan mata kuliah Sistem Kendali. Untuk memenuhi mata kuliah Motor Listrik kami membuat sistem magnetic stirrer dengan motor DC 3-6V dan dapat diatur kecepatannya sesuai dengan yang diinginkan. Sedangkan, untuk memenuhi mata kuliah Mikrokontroler kami menggunakan mikrokontroler jenis Arduino Uno R3 ATmega328P sebagai pengontrol sistem.

5.2 Saran

Untuk menyempurnakan sistem yang sudah kami buat, kami menyarankan:

1. Jika memungkinkan, element pemanas dibuat menjadi satu dengan dispenser.
2. Bracket untuk sensor, motor DC, tempat untuk menaruh dispenser, dan wadah untuk magnetic stirrer dapat dibuat dengan 3D agar lebih rapih, presisi, dan kokoh.
3. Sensor LDR diganti dengan sensor Load cell agar katup dispenser tidak mudah terbuka.

DAFTAR PUSTAKA

- Feri Djuandi. (2011). Pengenalan Arduino. *E-Book. Www. Tobuku*, 1–24.
<http://www.tobuku.com/docs/Arduino-Pengenalan.pdf>
- Golnaraghi, Fardi. Kuo, B. C. (2010). *Ninth Edition Farid Golnaraghi • Benjamin C. Kuo*.
- Hilal, A., & Manan, S. (2015). Pemanfaatan Motor Servo Sebagai Penggerak Cctv Untuk Melihat Alat-Alat Monitor Dan Kondisi Pasien Di Ruang Icu. *Gema Teknologi*, 17(2), 95–99. <https://doi.org/10.14710/gt.v17i2.8924>
- Setiawan, D., Yos Sudarso Km, J., Kunci, K., & Uno, A. (2017). Sistem Kontrol Motor Dc Menggunakan Pwm Arduino Berbasis Android System. *Jurnal Sains, Teknologi Dan Industri*, 15(1), 7–14.

LAMPIRAN

```
#include <Servo.h>

#include <Wire.h>

#include <Adafruit_MLX90614.h>

Adafruit_MLX90614 mlx = Adafruit_MLX90614();

String incomingByte;

bool stringComplete = false;

bool relayStatus = false;

bool stirrerStatus = false;

bool heaterStatus = false;

int pwm = 170;

int suhu = 0;

// Pin relay

const int relayPin = 7;

// Ambang batas suhu

float suhuOn = suhu - 1; // Relay menyala jika suhu >= 30°C

float suhuOff = suhu + 1; // Relay mati jika suhu < 28°C

unsigned long levelAir = 0;

Servo myservo;
```

```
int ldrPin = A0;

int ldrState = 0;

const int IN1 = 11;

const int IN2 = 10;

int motorState = 0;

int motorOn = 0;

bool servoState = false;

unsigned long servoStartTime = 0;

bool servoTimerActive = false;

unsigned long startTime = 0;

unsigned long previousMillis = 0; // Menyimpan waktu sebelumnya

const unsigned long interval = 100; // Interval waktu dalam milidetik (1 detik)

void setup() {

    pinMode(ldrPin, INPUT);

    pinMode(IN1, OUTPUT);

    pinMode(IN2, OUTPUT);

    mlx.begin();

    // Atur pin relay sebagai output

    pinMode(relayPin, OUTPUT);
```

```

digitalWrite(relayPin, LOW); // Pastikan relay mati saat awal

myservo.attach(8);

myservo.write(0);

Serial.begin(9600);
}

void loop() {
    unsigned long currentMillis = millis();

    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;

        if (stringComplete) {
            char prefix = incomingByte.charAt(0); // Ambil prefix ('S' atau 'P')

            String valueString = incomingByte.substring(1); // Ambil nilai angka

            int value = valueString.toInt(); // Konversi nilai ke integer

            if (prefix == 's') {
                // Proses data suhu

                suhu = value;

                suhuOn = suhu - 1; // Perbarui suhuOn

                suhuOff = suhu + 1;

                // Serial.print("Set temperature to: ");

```

```

// Serial.println(value);

// Tambahkan logika untuk mengatur suhu
} else if (prefix == 'p') {

// Proses data PWM

pwm = value;

// Serial.print("Set PWM to: ");

// Serial.println(value);

// Tambahkan logika untuk mengatur PWM
} else if (prefix == 'l') {

if(value == 200){

    levelAir = 18000;

}else if(value == 400){

    levelAir = 38000;

}else if(value == 600){

    levelAir = 57000;

}else {

    levelAir = 0;

}

Serial.println(levelAir);

}

if ( incomingByte == "H1") {

    relayStatus = true; // Aktifkan relay

```

```

        // Serial.println("Relay set to ON");

    }else if(incomingByte == "H0") {

        relayStatus = false; // Matikan relay

        //Serial.println("Relay set to OFF");

    }


    if ( incomingByte == "S1") {

        stirrerStatus = true; // Aktifkan relay

        // Serial.println("Relay set to ON");

    }else if(incomingByte == "S0") {

        stirrerStatus = false; // Matikan relay

        //Serial.println("Relay set to OFF");

    }


    // Reset input string

    incomingByte = "";

    stringComplete = false;

}


float suhuObjek = mlx.readObjectTempC();

ldrState = analogRead(ldrPin);

```



```

//Tampilkan suhu di serial monitor

// Serial.print("Suhu Objek: ");

Serial.println("c" + String(suhuObjek, 2));

// Serial.println(" °C");

// Serial.print("ldr: ");

// Serial.println(ldrState);

//Logika menyalakan dan mematikan relay

// Logika menyalakan dan mematikan relay


if (relayStatus) {

    if (suhuObjek <= suhuOn) {

        digitalWrite(relayPin, HIGH); // Relay menyala

        if(!heaterStatus){

            heaterStatus = true;

            Serial.println("h1");

        }

        // Serial.println("Relay ON: Suhu mendekati suhuOn");

    } else if (suhuObjek >= suhuOff) {

        digitalWrite(relayPin, LOW); // Relay mati

        if(heaterStatus){

            heaterStatus = false;

            Serial.println("h0");

        }

    }

}

```

```

    }

    // Serial.println("Relay OFF: Suhu mencapai suhuOff");

    }

} else {

    digitalWrite(relayPin, LOW); // Relay mati

    if(heaterStatus){

        heaterStatus = false;

        Serial.println("h0");

    }

}

if (ldrState <= 200 && !servoState && !servoTimerActive && stirrerStatus)
{ // Jika cahaya redup dan servo belum aktif

    servoState = true; // Tandai servo sebagai aktif

    servoTimerActive = true; // Aktifkan timer untuk servo

    servoStartTime = millis(); // Catat waktu servo aktif

    //Serial.println("Cahaya redup: Servo aktif (90 derajat).");

}

if(servoTimerActive && (millis() - servoStartTime >= 3000) && servoState
&& stirrerStatus){

    myservo.write(180);

```

```

    }

    // Periksa apakah 10 detik telah berlalu untuk servo

    if (servoTimerActive && (millis() - servoStartTime >= levelAir + 3000) ||
!stirrerStatus) {

        // Servo kembali ke posisi awal (tertutup)

        myservo.write(0);

        servoTimerActive = false; // Nonaktifkan timer untuk servo

        //Serial.println("Servo kembali ke posisi awal (tertutup). Motor akan mulai
berjalan.");

    }

    // Jika servo telah selesai tugasnya, jalankan motor

    if (!servoTimerActive && servoState) {

        if(stirrerStatus){

            if (motorState == 0) {

                motorState = 1; // Tandai motor sebagai aktif

                startTime = millis(); // Catat waktu motor mulai berjalan

            }

            // Jalankan motor dengan kecepatan tertentu

            maju(pwm, 170, 2000, 6000); // Normal=200, Pelan=170, Mati=5000ms,
Pelan=10000ms

        }

    }

```

```

// Jika cahaya terang, hentikan motor dan reset servo

if (ldrState > 200 || !stirrerStatus) {

    if (servoState) {

        myservo.write(0); // Pastikan servo tertutup

        servoState = false; // Reset status servo

        servoTimerActive = false; // Matikan timer servo

        //Serial.println("Cahaya terang: Servo dinonaktifkan.");

    }


    if (motorState == 1) {

        motorState = 0; // Reset status motor

        berhenti(100); // Hentikan motor

        //Serial.println("Cahaya terang: Motor berhenti.");

    }

}

}

void maju(int kecepatanNormal, int kecepatanPelan, unsigned long durasiMati,
unsigned long durasiPelan) {

    unsigned long elapsedTime = millis() - startTime;

```

```

if (elapsedTime <= durasiMati) {

    analogWrite(IN1, 0);

    analogWrite(IN2, 0);

    //Serial.println("Motor mati sementara.");

}else if (elapsedTime <= durasiPelan && elapsedTime > durasiMati) {

    analogWrite(IN1, kecepatanPelan);

    analogWrite(IN2, 0);

    // Serial.print("Fungsi maju dipanggil (kecepatan pelan)");

    // Serial.println(kecepatanPelan);

    if (motorOn == 0){

        motorOn = 1;

        Serial.println("m1");

    }

} else {

    analogWrite(IN1, kecepatanNormal);

    analogWrite(IN2, 0);

    // Serial.print("Fungsi maju dipanggil (kecepatan normal)");

    // Serial.println(kecepatanNormal);

}

}

void berhenti(int kecepatan) {

```

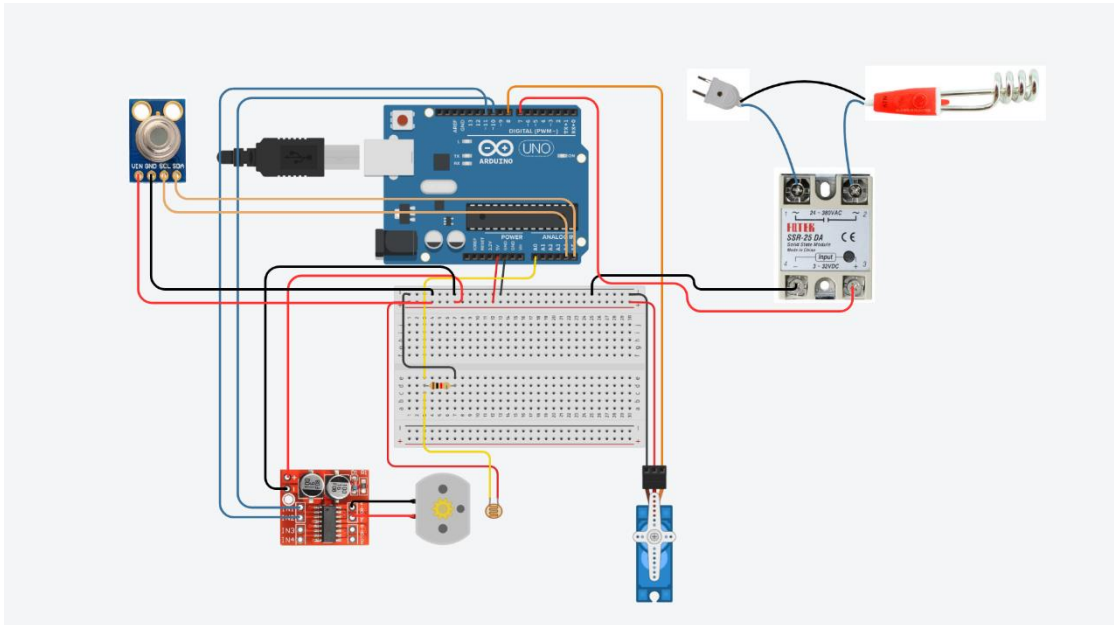
```

//Serial.println("Motor berhenti.");

analogWrite(IN1, 0);
analogWrite(IN2, kecepatan);
delay(500);
analogWrite(IN1, 0);
analogWrite(IN2, 0);
if (motorOn == 1){
    motorOn = 0;
    Serial.println("m0");
}
}

void serialEvent() {
    while (Serial.available()) {
        char inChar = (char)Serial.read();
        if (inChar == '\n') {
            stringComplete = true;
            break;
        } else {
            incomingByte += inChar;
        }
    }
}
}

```



Gambar 8. Wiring diagram sistem water heater dan magnetic stirrer