

**LAPORAN PRAKTIKUM
SISTEM PENGENDALIAN SUHU OTOMATIS
MATA KULIAH PRAKTIK AUTOMATION**



Disusun Oleh:

ANNISA ALLIYAH
2305112047

DOSEN PENGAMPU : AMIR SALEH, S.Pd., M.Kom.

**PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA
POLITEKNIK NEGERI MEDAN
MEDAN 2025**

DAFTAR ISI

BAB I PENDAHULUAN.....	3
1.1 Latar Belakang.....	3
1.2 Tujuan.....	3
BAB II DASAR TEORI.....	4
2.1 Alat Dan Bahan.....	4
2.1.1 Sensor Suhu DHT 22	4
2.1.2 Arduino Uno	4
2.1.3 Relay 2 Chanel.....	5
2.1.4 Kipas Mini	Error! Bookmark not defined.
2.1.5 Bread Board	5
2.1.6 Kabel Jumper	6
2.2 Cara Kerja Sistem.....	7
BAB III METODE PRAKTIKUM.....	8
3.2 Implementasi Dan Pengujian.....	8
3.2.1 Implementasi	8
3.2.2 Kode Program	9
3.2.3 Pengujian	10
3.3 Hasil Dan Diskusi.....	10
3.3.1 Hasil Pengujian	10
3.3.2 Diskusi.....	11
BAB IV KESIMPULAN.....	12
4.1 Kesimpulan.....	12

BAB I

PENDAHULUAN

1.1 Latar Belakang

Suhu merupakan salah satu faktor penting yang harus dijaga dalam berbagai lingkungan, seperti ruang server, rumah kaca, ruang penyimpanan makanan, maupun laboratorium. Perubahan suhu yang tidak terkendali dapat mengganggu kinerja perangkat, mempercepat kerusakan bahan, atau menurunkan kualitas hasil. Oleh karena itu, sistem pengendalian suhu otomatis sangat dibutuhkan agar suhu tetap stabil sesuai dengan batas yang telah ditentukan.

Dengan memanfaatkan sensor suhu, mikrokontroler, dan aktuator seperti kipas atau pemanas, sistem ini mampu mendeteksi perubahan suhu dan menyesuaikan secara otomatis tanpa intervensi manusia.

1.2 Tujuan

- Menjaga suhu pada level tertentu sesuai set point.
- Mengurangi intervensi manual dalam mengatur suhu.
- Meningkatkan efisiensi dan akurasi pengendalian suhu.

BAB II DASAR TEORI

2.1 Alat Dan Bahan

2.1.1 Sensor Suhu DHT 22



Gambar 2.1.1 ESP32..

Mendeteksi suhu lingkungan dan mengirimkan data ke mikrokontroler.

2.1.2 Sensor Ultrasonik



Gambar 2.1.2 Sensor Ultrasonik.

Memproses data dari sensor dan memberikan instruksi kepada aktuator.

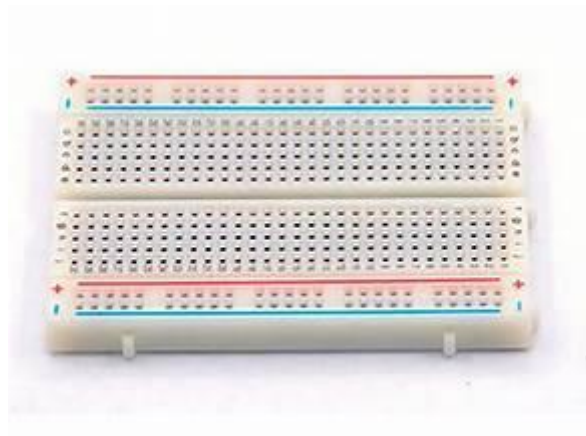
2.1.3 Servo



Gambar 2.1.3 servo

Berfungsi sebagai saklar elektronik untuk mengendalikan aktuator (kipas/pemanas) berdasarkan perintah dari Arduino.

2.1.4 Bread Board



Gambar 2.1.4 Bread Board

Breadboard adalah papan percobaan elektronik yang digunakan untuk menyusun dan menguji rangkaian tanpa perlu menyolder. Komponennya tinggal ditancapkan pada lubang-lubang konduktif di dalamnya, sehingga mudah dipasang, dipindahkan, atau dilepas.

2.1.5 Kabel Jumper



Gambar 2.1.5 Kabel Jumper

Kabel jumper adalah kabel kecil dengan ujung pin (male atau female) yang digunakan untuk menghubungkan komponen di breadboard atau dari breadboard ke Arduino/mikrokontroler.

2.2 Cara Kerja Sistem

- **Proses Deteksi Suhu:**

Sensor suhu mengukur suhu lingkungan dan mengirimkan data ke mikrokontroler.

- **Pemrosesan Data:**

Mikrokontroler memproses data suhu dan menentukan apakah perlu menyesuaikan suhu berdasarkan set point yang diinginkan.

- **Pengaturan Suhu:**

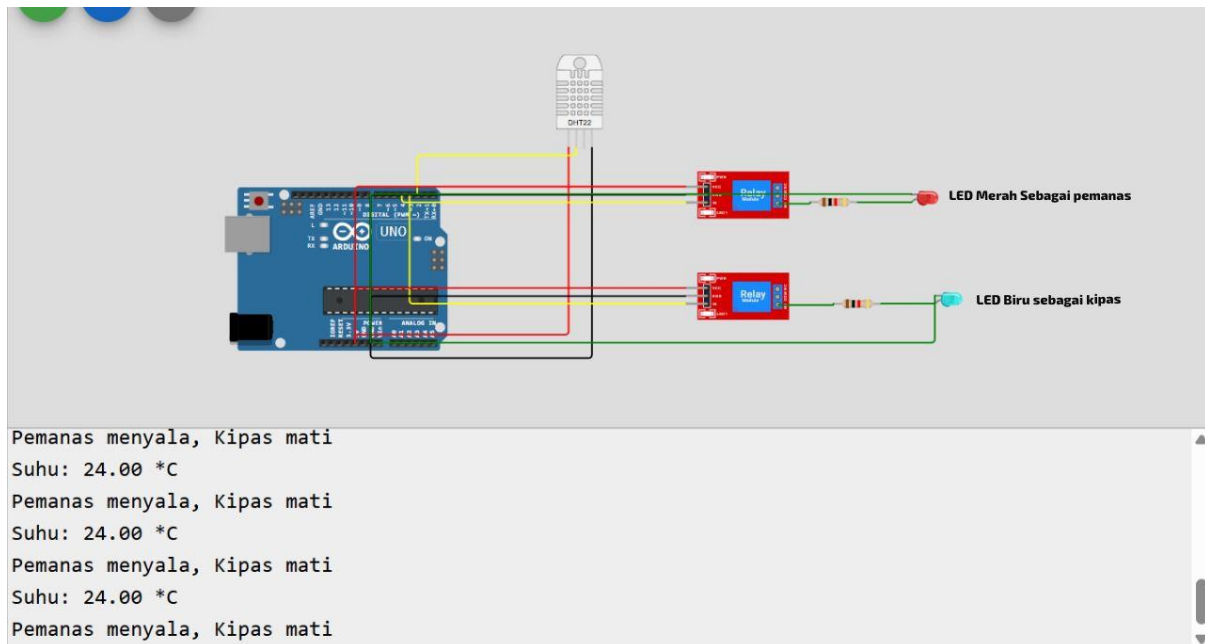
Jika suhu terlalu tinggi atau rendah, mikrokontroler akan mengaktifkan aktuator (misalnya, menghidupkan kipas jika suhu terlalu panas atau pemanas jika terlalu dingin) untuk menstabilkan suhu.

BAB III

METODE PRAKTIKUM

3.2 Implementasi Dan Pengujian

3.2.1 Implementasi



Gambar 3.2 Implementasi Rangkaian Pemasangan Komponen.

Sistem dirancang menggunakan:

- **Arduino Uno** sebagai mikrokontroler.
- **Sensor DHT22** untuk mendeteksi suhu lingkungan.
- **Kipas DC** sebagai aktuator pendingin.
- **Pemanas (simulasi: LED atau heater kecil)** sebagai aktuator penghangat.
- **Breadboard** dan **kabel jumper** untuk merangkai komponen.
- **Power supply/USB Arduino** untuk sumber daya.

Rangkaian disusun dengan detail berikut:

1. Pin data sensor DHT22 dihubungkan ke pin digital Arduino (misalnya D2).
2. Kipas DC dan pemanas dikendalikan melalui modul relay yang terhubung ke pin digital Arduino.
3. Breadboard digunakan sebagai media penyusunan rangkaian.
4. Kabel jumper digunakan untuk menghubungkan antar komponen.

3.2.2 Kode Program

```
#include "DHT.h"

#define DHTPIN 2      // Pin data DHT22
#define DHTTYPE DHT22 // Tipe sensor
DHT dht(DHTPIN, DHTTYPE);

#define FAN 3      // Relay kipas (LED biru)
#define HEATER 4    // Relay pemanas (LED merah)
#define SETPOINT 28 // Suhu target (°C)

void setup() {
    Serial.begin(9600);
    dht.begin();
    pinMode(FAN, OUTPUT);
    pinMode(HEATER, OUTPUT);
    digitalWrite(FAN, LOW);
    digitalWrite(HEATER, LOW);
}

void loop() {
    float suhu = dht.readTemperature();

    if (isnan(suhu)) {
        Serial.println("Gagal membaca sensor!");
        return;
    }

    Serial.print("Suhu: ");
    Serial.print(suhu);
    Serial.println(" *C");

    if (suhu > SETPOINT) {
        digitalWrite(FAN, HIGH); // Kipas ON
        digitalWrite(HEATER, LOW); // Pemanas OFF
        Serial.println("Kipas menyala, Pemanas mati");
    }
    else if (suhu < SETPOINT) {
        digitalWrite(FAN, LOW); // Kipas OFF
        digitalWrite(HEATER, HIGH); // Pemanas ON
        Serial.println("Pemanas menyala, Kipas mati");
    }
    else {
        digitalWrite(FAN, LOW);
        digitalWrite(HEATER, LOW);
    }
}
```

```

    Serial.println("Suhu stabil, semua OFF");
}

delay(2000); // baca setiap 2 detik
}

```

3.2.3 Pengujian

Pengujian dilakukan dengan mencoba sistem pada beberapa kondisi suhu:

1. **Suhu lebih tinggi dari set point (misalnya $> 28^{\circ}\text{C}$):**
 - Arduino mengaktifkan **kipas DC** untuk menurunkan suhu.
 - Pemanas tetap mati.
2. **Suhu lebih rendah dari set point (misalnya $< 24^{\circ}\text{C}$):**
 - Arduino mengaktifkan **pemanas** untuk menaikkan suhu.
 - Kipas tetap mati.
3. **Suhu dalam rentang normal ($24\text{--}28^{\circ}\text{C}$):**
 - Baik kipas maupun pemanas **tidak aktif**, karena suhu sudah stabil.

Hasil pengujian menunjukkan sistem dapat menjaga suhu sesuai dengan set point, meskipun ada sedikit keterlambatan respon (delay) antara sensor membaca suhu dan aktuator bekerja.

3.3 Hasil Dan Diskusi

3.3.1 Hasil Pengujian

Pengujian dilakukan dengan memberikan variasi suhu pada sensor DHT22, hasilnya sebagai berikut:

Kondisi Suhu ($^{\circ}\text{C}$)	Status Kipas	Status Pemanas	Keterangan
22	Mati	Menyala	Suhu terlalu dingin, pemanas aktif
25	Mati	Mati	Suhu dalam rentang normal
28	Mati	Mati	Suhu dalam rentang normal
31	Menyala	Mati	Suhu terlalu panas, kipas aktif

Dari tabel terlihat sistem bekerja sesuai dengan logika yang diprogram:

- Jika suhu **terlalu rendah** → pemanas aktif.
- Jika suhu **terlalu tinggi** → kipas aktif.
- Jika suhu **normal** → aktuator mati.

3.3.2 Diskusi

- **Keefektifan Sistem**

- Sistem berhasil menjaga suhu dalam rentang yang diinginkan dengan respon otomatis.
- Kinerja sensor DHT22 cukup akurat dalam membaca suhu.
- Aktuator (kipas/pemanas) merespons sesuai dengan perintah Arduino.

- **Kelemahan Sistem**

- Respon perubahan suhu tidak instan karena dipengaruhi oleh **delay sensor** dan **inertia suhu ruangan**.
- Pemanas yang digunakan masih berupa simulasi (misalnya LED atau heater kecil), belum mampu menaikkan suhu ruangan besar.
- Konsumsi daya bisa meningkat jika kipas atau pemanas bekerja terlalu lama.

- **Potensi Pengembangan Lebih Lanjut**

- Menambahkan **tampilan LCD/OLED** untuk memonitor suhu secara real-time.
- Menambahkan **kontrol berbasis IoT (misalnya via aplikasi/web)** agar suhu bisa dipantau dari jarak jauh.
- Menggunakan **aktuator yang lebih efisien**, misalnya kipas hemat daya atau pemanas dengan pengendalian daya variabel (PWM).
- Menerapkan **algoritma kontrol PID** agar respon sistem lebih halus dan stabil.

BAB IV

KESIMPULAN

4.1 Kesimpulan

- Sistem pengendalian suhu otomatis berbasis **Arduino Uno dan sensor DHT22** mampu menjaga suhu pada rentang yang ditentukan dengan mengaktifkan **kipas** saat suhu terlalu panas dan **pemanas** saat suhu terlalu dingin.
- Pengujian menunjukkan sistem bekerja sesuai logika yang diprogram, meskipun terdapat sedikit keterlambatan respon.
- Sistem ini cukup efektif digunakan untuk skala kecil, namun untuk aplikasi nyata dalam ruangan besar diperlukan aktuator yang lebih kuat.
- Pengembangan lebih lanjut dapat dilakukan dengan menambahkan **tampilan monitoring (LCD/IoT)** serta menggunakan metode kontrol yang lebih canggih seperti **PID control** untuk menjaga kestabilan suhu.