

**LAPORAN PRAKTIKUM
KONTROL PROSES**



DISUSUN OLEH:

NAMA : M. Yusuf Riyadi
NIM : 40040323620003
KELAS : A

**PROGRAM STUDI STr. TEKNOLOGI REKAYASA OTOMASI
SEKOLAH VOKASI - UNIVERSITAS DIPONEGORO
TAHUN 2026**

MODUL 1

SISTEM KONTROL TEMPERATUR ON-OFF

I. Langkah Percobaan

1. Kabel daya (power) modul alat praktikum dihubungkan ke sumber tegangan listrik.
2. Lampu pijar (sebagai aktuator sumber panas) diposisikan dan direkatkan agar berdekatan dengan sensor suhu DHT untuk mengoptimalkan pembacaan suhu dan respons sistem.
3. Board mikrokontroler pada alat dihubungkan ke laptop menggunakan kabel USB.
4. Aplikasi antarmuka (HMI/GUI) kontrol suhu pada laptop dibuka dan dijalankan.
5. Konfigurasi komunikasi serial dilakukan dengan menekan menu "Set Serial". Pada jendela Setup, parameter berikut disesuaikan:
 - Port: (COM dipilih sesuai dengan koneksi mikrokontroler)
 - Baud rate: 9600
 - Data bits: 8
 - Stop bits: 1
 - Parity: None
 - Flow control: NoneSetelah itu, tombol OK diklik.
6. Tombol "Connect" pada aplikasi diklik untuk membangun komunikasi antara antarmuka di laptop dengan perangkat keras mikrokontroler.
7. Nilai batas suhu yang diinginkan dimasukkan pada kolom Set Point.
8. Tombol "Start" ditekan untuk memulai proses pengendalian. Lampu akan secara otomatis dinyalakan oleh alat untuk menaikkan suhu hingga mencapai nilai Set Point, dan akan dimatikan ketika suhu tersebut sudah tercapai (On-Off Control).
9. Perubahan suhu pada grafik antarmuka diamati, data yang dihasilkan dicatat, dan dokumentasi diambil sebagai bukti hasil percobaan.

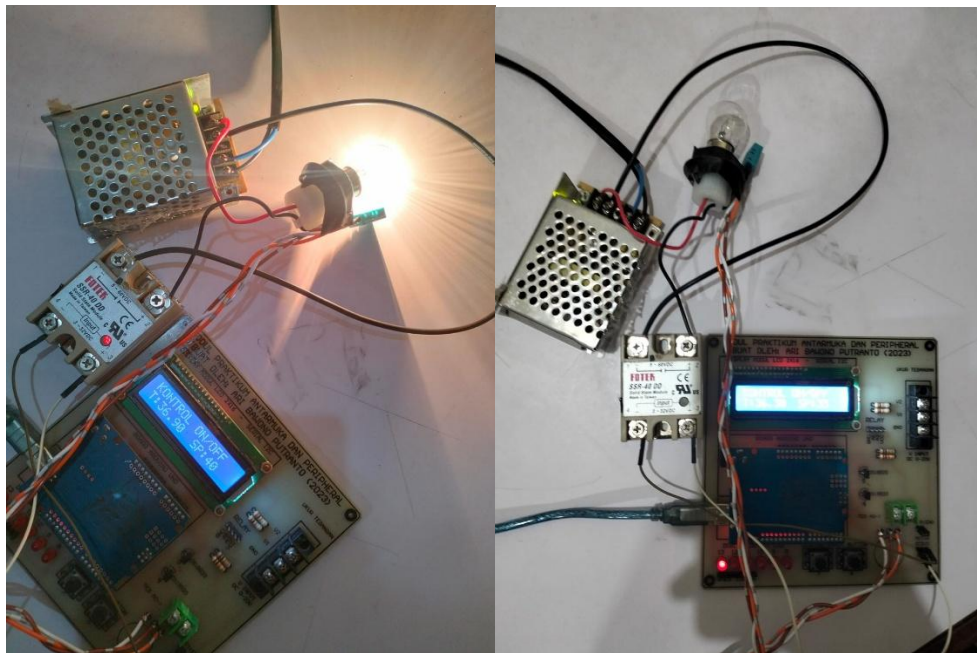
10. Langkah 7 hingga 9 diulangi untuk melakukan pengambilan data. Pada modul ini, percobaan dilakukan sebanyak tiga kali menggunakan tiga variasi nilai Set Point suhu yang berbeda, yaitu 35°C, 40°C, dan 50°C.

II. Hasil Percobaan

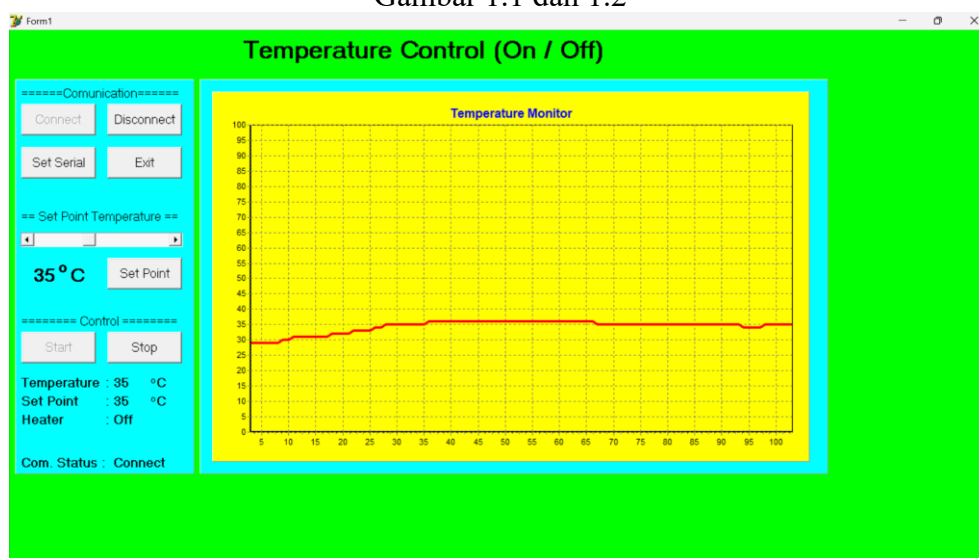
Pada bagian ini, disajikan data hasil pengujian sistem kendali temperatur siklus tertutup menggunakan metode kontrol ON-OFF. Hasil pengujian dibagi menjadi dua representasi visual, yaitu dokumentasi konfigurasi perangkat keras (*hardware*) saat sistem beroperasi, dan perekaman grafik respons suhu melalui aplikasi antarmuka (*Human-Machine Interface* / HMI) pada PC.

Gambar 1.1 dan Gambar 1.2 menunjukkan implementasi rangkaian alat yang terdiri dari mikrokontroler, modul *Solid State Relay* (SSR), sensor DHT22, serta lampu pijar sebagai elemen pemanas. Dokumentasi tersebut memperlihatkan kondisi sistem saat aktuator menyala (mengejar target suhu) dan saat aktuator padam.

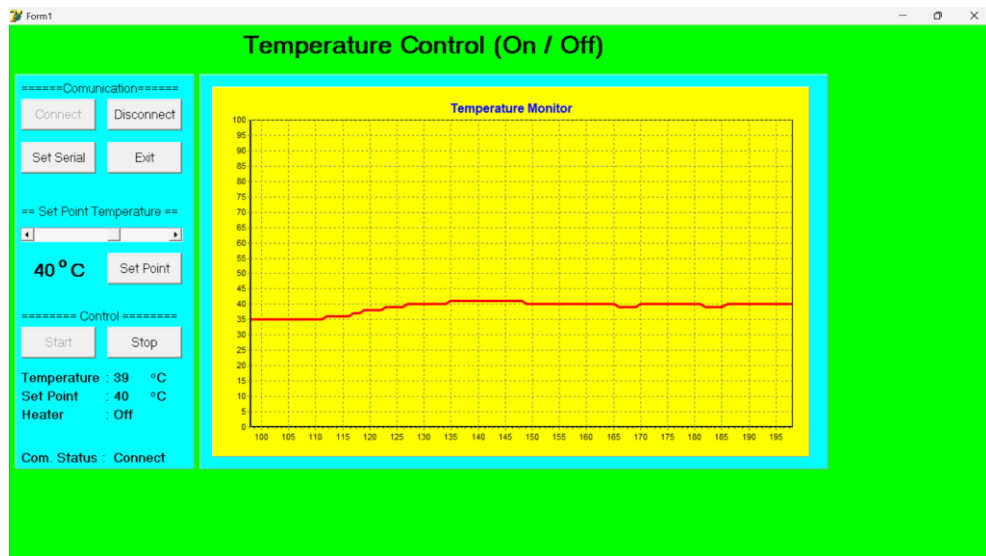
Selanjutnya, dinamika perubahan suhu aktual terhadap waktu direkam dan divisualisasikan melalui aplikasi HMI. Pengujian dilakukan dengan memberikan tiga variasi *Set Point* yang berbeda. Gambar 1.3 menampilkan grafik pemantauan saat *Set Point* diatur pada 35°C, Gambar 1.4 pada target 40°C, dan Gambar 1.5 menunjukkan respons sistem pada target tertinggi yaitu 50°C. Grafik-grafik ini mempresentasikan profil *rise time* serta osilasi suhu yang menjadi karakteristik utama dari pengendalian ON-OFF.



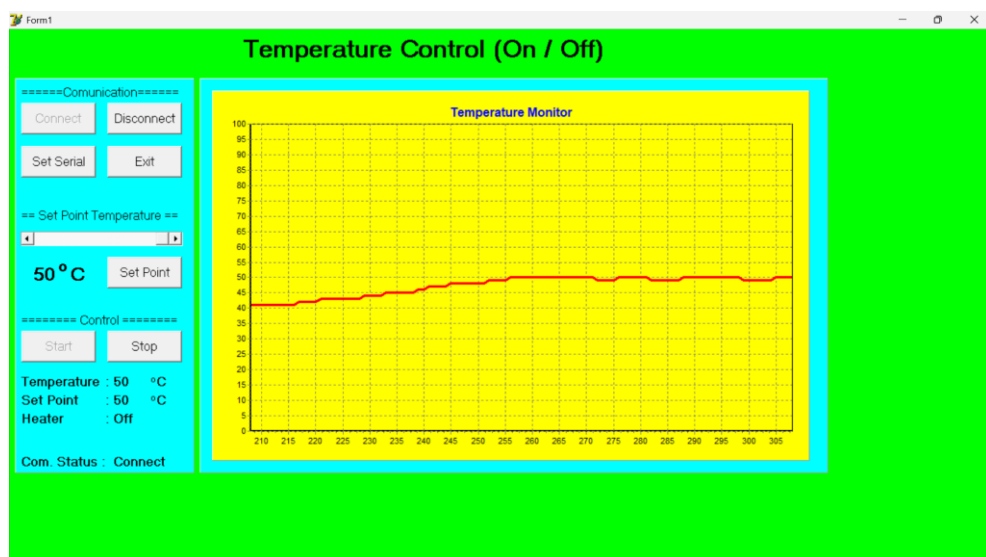
Gambar 1.1 dan 1.2



Gambar 1.3



Gambar 1.4



Gambar 1.5

III. Listing Program Mikrokontroler

```
#include "DHT.h"           // Memasukkan library untuk sensor suhu dan kelembaban
DHT                         DHT

#define DHTPIN 2            // Mendefinisikan pin digital 2 pada Arduino sebagai
pin data sensor DHT
#define DHTTYPE DHT22      // Mendefinisikan tipe sensor DHT yang digunakan, yaitu
DHT22

// Membuat objek 'dht' dengan parameter pin dan tipe sensor yang telah
didefinisikan sebelumnya
DHT dht(DHTPIN, DHTTYPE);

// --- Deklarasi Variabel Global ---
```

```

int set_point = 30;           // Variabel untuk menyimpan nilai batas suhu (set
point), nilai awal 30
String data_masuk;           // Variabel bertipe String untuk menyimpan data yang
diterima dari komunikasi serial (dari HMI/Laptop)
int kirim;                   // (Variabel ini dideklarasikan namun tidak digunakan
dalam logika program ini)
int looping = 0;             // Variabel status (flag) untuk mengontrol jalannya
sistem. 0 = Sistem Berhenti (Stop), 1 = Sistem Berjalan (Start)

void setup() {
    pinMode(3, OUTPUT);       // Mengatur pin digital 3 sebagai OUTPUT (dihubungkan
ke Solid State Relay/SSR untuk kontrol lampu pemanas)
    digitalWrite(3, LOW);     // Memberikan logika LOW pada awal program agar
lampu/pemanas dalam kondisi mati (OFF) saat alat pertama kali dinyalakan

    Serial.begin(9600);       // Memulai komunikasi serial dengan baud rate 9600 bps
(harus sama dengan pengaturan di aplikasi HMI)
    dht.begin();              // Menginisialisasi atau memulai pembacaan sensor DHT
}

void loop() {
    delay(1000);              // Memberikan jeda waktu 1 detik (1000 ms) di setiap
perulangan agar sensor DHT memiliki waktu untuk stabil membaca data

    float suhu = dht.readTemperature(); // Membaca data suhu (dalam derajat Celcius)
dari sensor dan menyimpannya di variabel 'suhu'

    // --- Blok Komunikasi Serial (Menerima Perintah dari HMI) ---
    if (Serial.available()) // Mengecek apakah ada data yang masuk ke buffer serial
dari laptop/HMI
    {
        data_masuk = Serial.readString(); // Membaca seluruh data string yang masuk
dan menyimpannya di variabel 'data_masuk'

        // Jika karakter pertama dari data yang masuk adalah huruf "A" (Perintah START
dari HMI)
        if (data_masuk.substring(0,1) == "A")
        {
            looping = 1;        // Mengubah status sistem menjadi berjalan (aktif)
        }

        // Jika karakter pertama dari data yang masuk adalah huruf "B" (Perintah STOP
dari HMI)
        if (data_masuk.substring(0,1) == "B")
        {
            looping = 0;        // Mengubah status sistem menjadi berhenti (non-aktif)
            digitalWrite(3, LOW); // Mematikan lampu pemanas seketika demi keamanan saat
sistem di-stop
        }

        // Jika data yang masuk BUKAN "A" dan BUKAN "B" (Diasumsikan sebagai data nilai
Set Point baru dari HMI)
        if ((data_masuk.substring(0,1) != "A") && (data_masuk.substring(0,1) != "B"))
        {
            set_point = (data_masuk.toInt()); // Mengubah data string menjadi integer
(angka) dan menyimpannya sebagai nilai set point baru
        }
    }

    // --- Blok Pengendalian On-Off (Hanya berjalan jika status 'looping' = 1 /
tombol Start sudah ditekan) ---

```

```

if (looping == 1)
{
    // Mengirim data suhu ke HMI
    // Suhu dikalikan 100 dan di-casting menjadi integer (misal 35.50 menjadi 3550)
    // untuk memudahkan pengiriman serial
    Serial.print((int)suhu * 100);
    Serial.println('X');    // Mengirimkan karakter 'X' sebagai penanda/pembatas
    // akhir data suhu untuk dibaca oleh HMI

    // Mengirim data set point ke HMI
    Serial.print(set_point);
    Serial.println('Y');    // Mengirimkan karakter 'Y' sebagai penanda/pembatas
    // akhir data set point untuk dibaca oleh HMI

    // Logika Kontrol ON-OFF
    if (suhu >= set_point)
    {
        digitalWrite(3, LOW); // Jika suhu saat ini LEBIH BESAR atau SAMA DENGAN set
        // point, matikan pemanas (SSR OFF)
    }
    if (suhu < set_point)
    {
        digitalWrite(3, HIGH); // Jika suhu saat ini LEBIH KECIL dari set point,
        // nyalakan pemanas (SSR ON) untuk menaikkan suhu
    }
}
}

```

IV. Pembahasan Hasil Percobaan

Praktikum ini mengimplementasikan sistem kendali suhu siklus tertutup (*closed-loop control system*) dengan menggunakan metode aksi pengontrolan dua posisi (*ON-OFF Control / Two-Position Control*). Sistem ini mengintegrasikan mikrokontroler sebagai pusat pemrosesan logika, sensor DHT22 sebagai elemen umpan balik (*feedback*), *Solid State Relay* (SSR) sebagai aktuator peralihan, lampu pijar sebagai *plant* (sumber panas), serta HMI berbasis PC untuk *monitoring* dan *supervisory control*.

Berdasarkan data yang ditarik dari antarmuka (*software*) dan observasi perangkat keras (*hardware*), berikut adalah analisis menyeluruh terhadap sistem:

1. Analisis Kinerja Pengendalian Berdasarkan Variasi *Set Point*

Sistem diuji dengan memberikan tiga nilai target suhu (*Set Point*) yang berbeda untuk mengamati profil respons waktu dan kestabilan termal:

- *Set Point* 35°C: Berdasarkan dokumentasi perangkat keras, ketika suhu aktual terukur 36.30°C (melebihi batas *Set Point* 35°C), kontroler merespons dengan tepat dengan memberikan sinyal LOW ke pin aktuator.

SSR terputus, dan lampu pijar padam. Grafik pada HMI menunjukkan sistem dengan cepat mencapai area 35°C, namun fluktuasi (*ripple*) terjadi dengan frekuensi yang cukup rapat di sekitar garis target karena suhu ruangan secara alami cenderung menarik panas tersebut turun dengan cepat.

- *Set Point* 40°C: Saat target dinaikkan ke 40°C, dokumentasi menunjukkan sistem memberikan sinyal HIGH karena suhu aktual baru mencapai 36.90°C (kondisi *error* bernilai positif / *Set Point* > Suhu Aktual). Lampu pijar menyala terang untuk melakukan injeksi panas. Pada HMI, kurva menunjukkan waktu naik (*rise time*) yang stabil. Sistem terus memaksa aktuator bekerja 100% (*full load*) hingga titik 40°C tertembus.
- *Set Point* 50°C: Pada pengujian tertinggi ini, kurva *monitoring* memperlihatkan waktu pencapaian target yang paling lambat. Hal ini dipengaruhi oleh kapasitas daya lampu pijar yang memiliki batas maksimal perpindahan panas termodinamika ke lingkungan sekitarnya. Saat mencapai rentang 50°C, grafik menunjukkan osilasi periodik yang sangat jelas, memvalidasi bahwa sistem secara konstan berjuang mempertahankan suhu terhadap laju pendinginan lingkungan (*heat loss*).

2. Analisis Karakteristik Kontrol ON-OFF (Analisis Kurva Transient & Steady-State)

Dari ketiga grafik yang dihasilkan pada HMI, karakteristik paling menonjol yang diamati adalah tidak adanya garis lurus sempurna saat suhu mencapai target. Selalu terdapat gelombang osilasi berkelanjutan (*sustained oscillation*). Fenomena ini disebabkan oleh dua faktor fisis:

- *Overshoot* (Suhu bablas ke atas): Diakibatkan oleh inersia termal pada aktuator. Ketika kontroler mendeteksi suhu sudah mencapai *Set Point* dan mematikan SSR, filamen dan kaca lampu pijar masih menyimpan panas laten. Panas sisa ini terus teradiasi ke sensor DHT22, menyebabkan grafik suhu tetap naik melebihi batas sesaat setelah lampu padam.
- *Undershoot* (Suhu turun terlalu jauh): Terjadi ketika lampu pijar dinyalakan kembali karena suhu jatuh di bawah *Set Point*. Lampu pijar membutuhkan

waktu (transien termal) untuk memanaskan filamennya kembali hingga titik di mana panasnya bisa dideteksi oleh sensor.

Kondisi *overshoot* dan *undershoot* ini dipertegas oleh logika di dalam *source code* (penggunaan `delay(1000);`). Jeda pengambilan sampel data selama 1 detik menciptakan area "buta" (*dead-time/delay time*) pada sistem kontrol, di mana perubahan suhu yang drastis di sepersekian detik tidak langsung terkompensasi oleh aktuator.

3. Kinerja Komunikasi Serial dan Periferal Perangkat Keras

- Stabilitas Komunikasi Data: Implementasi protokol komunikasi serial menggunakan pengurai karakter (*string parsing*) "A" untuk Start, "B" untuk Stop, dan pembatas "X" serta "Y" untuk pemisahan paket data suhu dan target. Konfigurasi COM Port dengan *Baud Rate* 9600 bps beroperasi secara asinkron tanpa *packet loss*. Data dapat direpresentasikan ke dalam bentuk grafik *real-time* di HMI tanpa adanya *lag* visual yang mengganggu.
- Keandalan Aktuator (*Solid State Relay / SSR*): Pemilihan SSR (Fotek SSR-40 DD) untuk mengontrol lampu pijar sangat efektif untuk sistem kendali ON-OFF yang membutuhkan peralihan status (ON/OFF) secara cepat dan terus-menerus. SSR beroperasi menggunakan *optocoupler* (isolasi cahaya) di dalamnya, sehingga tidak ada kontak mekanis. Hal ini mencegah terjadinya *bouncing* arus, loncatan bunga api (*sparking*), serta meniadakan keausan mekanik yang biasa terjadi jika menggunakan *relay* konvensional pada frekuensi pensaklaran tinggi.

V. Kesimpulan

Berdasarkan percobaan sistem kendali temperatur ON-OFF yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Keberhasilan Sistem Integrasi: Sistem kendali temperatur siklus tertutup (*closed-loop*) menggunakan metode kontrol ON-OFF telah berhasil diimplementasikan. Mikrokontroler, sensor DHT22, *Solid State Relay* (SSR), dan aktuator pemanas (lampu pijar) mampu bekerja dengan baik

serta terintegrasi secara *real-time* dengan perangkat lunak HMI di PC melalui komunikasi serial yang stabil.

2. Karakteristik Kontrol ON-OFF: Hasil pemantauan grafik HMI pada ketiga variasi *Set Point* (35°C, 40°C, dan 50°C) membuktikan karakteristik fundamental dari sistem kontrol ON-OFF, yaitu selalu terjadi osilasi (fluktuasi suhu) di sekitar titik target. Sistem tidak akan pernah mencapai kondisi *steady-state* (garis lurus sempurna) karena kendali hanya bekerja dalam dua status mutlak: mati (0%) atau menyala penuh (100%).
3. Pengaruh Inersia Termal (Overshoot & Undershoot): Sistem mengalami *overshoot* (suhu melewati target batas atas) akibat sisa radiasi panas laten dari kaca lampu pijar meskipun SSR telah dimatikan. Sebaliknya, *undershoot* (suhu turun melewati batas bawah target) terjadi karena laju pendinginan lingkungan dan waktu tunda (*delay*) elemen pemanas untuk kembali mencapai suhu optimal setelah SSR dinyalakan ulang.
4. Keandalan Periferal Perangkat Keras: Penggunaan *Solid State Relay* (SSR) sangat direkomendasikan dan terbukti andal untuk metode kontrol ON-OFF. Karena tidak menggunakan sakelar mekanis, SSR mampu mengakomodasi peralihan arus tegangan tinggi dengan cepat, aman, dan tanpa risiko keausan mekanik akibat *switching* yang terus-menerus.